UNIVERSIDAD DE VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

ANÁLISIS EMOCIONAL SOBRE CONTENIDO DE SALUD MENTAL EN INSTAGRAM MEDIANTE MODELOS BERT

Autor:
Dña. Lucía Diez Platero
Tutor:
Dña. Noemí Merayo Álvarez

TÍTULO: **Análisis emocional sobre contenido de salud mental en Instagram mediante modelos Bert**
AUTOR: **Dña. Lucía Diez Platero**
TUTOR: **Dña. Noemí Merayo Álvarez**
DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

TRIBUNAL
PRESIDENTE:
SECRETARIO:
VOCAL:
SUPLENTE:
SUPLENTE:

FECHA:
CALIFICACIÓN:

Resumen de TFG

Este trabajo se enfoca en el análisis de las respuestas emocionales en comentarios de publicaciones en redes sociales, centrándonos en Instagram, mediante técnicas de Inteligencia Artificial y Procesamiento de Lenguaje Natural.

En los últimos años, el uso masivo de las redes sociales ha tenido un impacto significativo en la sociedad, fomentando una mayor interacción por parte de los usuarios. Este fenómeno, junto con la creciente exposición de los *Influencer*s a temas relacionados con la salud mental, ha generado la necesidad de comprender las emociones que despiertan estas publicaciones. Por ello, dicho estudio se centra en optimizar modelos de clasificación tanto de polaridad (positiva, negativa o indeterminada/neutra) como de emociones (amor/admiración, gratitud, comprensión/identificación/empatía, tristeza/pena, enfado/desprecio/burla o indeterminado) para mejorar los resultados obtenidos en investigaciones anteriores y así poder analizar y estudiar el impacto en la sociedad con mayor precisión.

Para lograrlo, se ha ampliado un corpus de datos que servirá para el entrenamiento del modelo, asignando a cada comentario sus respectivas etiquetas de polaridad y emoción. A continuación, los mensajes extraídos del corpus se someten a un proceso de limpieza para eliminar información redundante y normalizar los datos. Posteriormente los comentarios se introducen en un modelo de clasificación basado en Transformadores específico para el lenguaje en español, cuyo fin es entrenarlo y optimizarlo.

Finalmente, este modelo se ha integrado en una interfaz que permite al usuario realizar predicciones cargando un fichero o introduciendo un comentario manualmente, mostrando los resultados mediante gráficas. Además, estos resultados pueden ser descargados en un fichero para su posterior análisis.

Palabras clave

Salud mental, sentimientos, emociones, polaridad, Instagram, BERT, interfaz.

Abstract

This work focuses on the analysis of emotional responses in comments on social media posts, focusing on Instagram, using Artificial Intelligence and Natural Language Processing techniques.

In recent years, the massive use of social media has had a significant impact on society, encouraging greater user interaction. This phenomenon, together with the growing exposure of *Influencer*s to mental health-related topics, has generated the need to understand the emotions aroused by these publications. Therefore, this study focuses on optimising classification models for both polarity (positive, negative or indeterminate/neutral) and emotions (love/admiration, gratitude, understanding/identification/empathy, sadness/pity, anger/disdain/ derision or indeterminate) to improve the results obtained in previous research and thus be able to analyse and study the impact on society with greater precision.

To achieve this, a corpus of data has been extended for training the model by assigning each comment its respective polarity and emotion labels. The messages extracted from the corpus are then subjected to a cleaning process to remove redundant information and normalise the data. Subsequently, the comments are introduced into a Transformer-based classification model specific to the Spanish language, in order to train and optimise it.

Finally, this model has been integrated into an interface that allows the user to make predictions by uploading a file or entering a comment manually, displaying the results in graphs. In addition, these results can be downloaded as a file for further analysis.

Keywords

Mental health, feelings, emotions, polarity, Instagram, BERT, interface.

Agradecimientos

Quiero empezar dedicando este trabajo a mi familia y amigos por apoyarme y sacar siempre lo mejor de mí. Sé que el abuelo está muy orgulloso y nos cuida desde arriba.

Pero en especial a mamá y a papá, gracias por quererme, cuidarme y confiar en mí cuando ni yo misma lo hacía. Sois mi referente y mi guía. Gracias por siempre darme alas.

Por último, agradecer a mi tutora Noemí el privilegio de haber podido compartir este proyecto juntas. Gracias por el corazón y la pasión que le pones a todo.

Índice

Índice de figuras

Índice de tablas

1

Introducción

1.1 Motivación

El presente Trabajo de Fin de Grado (TFG) se enfoca en el análisis de la respuesta emocional con técnicas de Inteligencia Artificial (IA) en el contexto de la salud mental en redes sociales, en concreto Instagram. El uso masivo de redes sociales tanto por jóvenes como por adultos las ha convertido en uno de los actuales medios de comunicación por excelencia. Por consiguiente, las pone en el punto de mira perfecto para utilizarlas como herramienta de análisis del estado de la salud mental de la población actual.

Otro factor tenido en cuenta para la realización de este proyecto es su posible aporte a la comunidad psicológica, ya que puede ayudar a los profesionales a analizar y evaluar cómo influyen las redes sociales en la salud mental de la sociedad [1].

1.2 Objetivos

1.2.1 Objetivo principal

El principal objetivo de este trabajo es utilizar técnicas de Inteligencia Artificial para analizar la respuesta emocional de comentarios extraídos de posts en Instagram que hablan sobre problemas de salud mental. Para ello nos centraremos en la clasificación de la polaridad (positiva, negativa e indeterminada/neutra) y las emociones (amor/admiración, gratitud, comprensión/identificación/empatía, tristeza/pena, enfado/desprecio/burla e indeterminado) que desprenden dichos comentarios.

Este trabajo parte de los resultados obtenidos de un estudio previo [2] que también aborda el análisis de la respuesta emocional en comentarios. Se pretende optimizar los resultados de dicho estudio ampliando el corpus de datos y ajustando los hiperparámetros del modelo RoBERTuito, un modelo de clasificación basado en BERT (Bidirectional

Encoder Representations from Transformers), adaptado al español y al entorno de redes sociales.

Finalmente, el objetivo es optimizar un modelo basado en la arquitectura BERT capaz de clasificar con la máxima precisión la polaridad y las emociones expresadas en los comentarios de los posts de Instagram seleccionados. Además, se integrará en una interfaz desde la cual cualquier usuario podrá utilizar esta herramienta fácilmente. Todo esto permitirá una mejor comprensión de la dinámica de las comunidades en Instagram y favorecerá el desarrollo de herramientas más avanzadas para el análisis de sentimientos en el ámbito de las redes sociales.

1.2.2 Objetivos específicos

Para lograr el objetivo de este TFG se han desarrollado los siguientes objetivos específicos:

1. Ampliación del corpus de datos de Instagram. Con el fin de equilibrar la cantidad de comentarios según emociones y polaridad, se ha puesto especial atención en aquellas categorías con menor representación en el corpus original, procurando así un mejor balance.
2. Búsqueda y ajuste de hiperparámetros óptimos para el modelo de clasificación BERT seleccionado (RoBERTuito) con el objetivo de maximizar su precisión en la detección de polaridad y emociones, empleando el corpus ampliado y equilibrado mencionado en el punto anterior.
3. Evaluación el rendimiento del modelo optimizado en la clasificación de polaridad y emociones utilizando métricas como precisión, exhaustividad, puntuación F1 y precisión global. Además, se compararán los resultados con los del estudio anterior para evaluar si se ha obtenido una mejora significativa.
4. Desarrollo de una interfaz gráfica que facilite la aplicación del modelo optimizado, permitiendo una interacción más intuitiva y visual, además de obtener resultados de forma inmediata.

1.3 Metodología de trabajo

La metodología seguida para alcanzar los objetivos mencionados anteriormente se ha estructurado en cuatro fases que desarrollamos a continuación.

1.3.1 Fase de documentación

Para esta primera fase ha sido fundamental establecer una base sólida para poder, una vez comprendidos los conceptos clave, realizar el estudio de los comentarios recogidos de publicaciones sobre salud mental en Instagram. Para poder seleccionar las técnicas y enfoques más adecuados ha sido necesario realizar una exploración en profundidad de las características de los modelos de lenguaje basados en Transformers. En nuestro caso nos hemos centrado en RoBERTuito, una variante de los modelos BERT para lenguaje en español y centrado en redes sociales.

Además, se han investigado herramientas y librerías de IA para facilitar la implementación, ajuste y análisis del modelo para posteriormente poder llevar a cabo su optimización.

1.3.2 Fase de análisis

Esta segunda fase se centró en la evaluación de diferentes estrategias para que la selección de los datos para nuestro modelo obtuviera la mayor precisión y rendimiento posibles. Para ello se analizaron los comentarios de múltiples posts recogidos de Instagram para equilibrar el corpus inicial centrándonos en aquellas emociones más difíciles de interpretar en el contexto de la salud mental, obteniendo así un corpus más equilibrado.

A su vez, se analizaron los resultados de trabajos anteriores [2,3] para obtener una idea más clara de que resultados esperar del modelo, centrándonos en las métricas de rendimiento y matrices de confusión. En concreto, nos apoyamos en los resultados del TFG de Javier Estévez Asensio [2] y del TFG de Miguel Carralero Lanchares [3].

1.3.3 Fase de prueba

Tercera fase del proyecto en la que nos dedicamos a la evaluación y optimización del modelo. Para ello entrenamos el modelo BERT seleccionado, RoBERTuito, con los datos procesados y los hiperparámetros óptimos. Una vez obtenidos los resultados de las métricas de rendimiento, pasamos a realizar una comparación con los trabajos

anteriormente mencionados. Evaluamos las similitudes y diferencias tanto de polaridad y de emociones para identificar si se ha realizado una mejoría.

1.3.4 Fase de escritura

Cuarta y última fase en la cual se ha redactado el presente Trabajo de Fin de Grado, detallando todo el proceso y explicando el resultado obtenido.

1.4 Estructura de la memoria

La memoria del proyecto se distribuye en varios capítulos, cada uno centrado en distintos objetivos:

Capítulo 1: Introducción, se presenta el problema a desarrollar, los objetivos a lograr y la metodología a seguir para llegar a la resolución.

Capítulo 2: Estado del arte, se brinda una breve explicación sobre modelos de aprendizaje automático y procesamiento del lenguaje natural centrándonos en la arquitectura *Transformers* y los modelos BERT para poner en contexto al lector y facilitar la lectura y comprensión del documento. Además, se examina una visión actual del impacto de las redes sociales y la salud mental.

Capítulo 3: Descripción de las principales herramientas software utilizadas para el estudio y su posterior análisis.

Capítulo 4: Ampliación del corpus inicial con comentarios obtenidos de publicaciones sobre salud mental en Instagram. Se detallan los criterios de selección de los *posts*, el proceso de etiquetado y ejemplos utilizados.

Capítulo 5: Análisis de la respuesta emocional empleando el corpus anteriormente mencionado utilizando el modelo RoBERTuito. Nos centraremos en las etapas de preprocesamiento de los datos, elección de hiperparámetros y entrenamiento del modelo. Finalmente realizaremos un análisis detallado de las métricas y resultados obtenidos.

Capítulo 6: Diseño y programación de la interfaz gráfica para poder integrar el modelo y, por tanto, ser utilizado por cualquier usuario.

Capítulo 7: Conclusiones del estudio y propuestas de líneas futuras de investigación.

2

Estado del arte

2.1 Introducción

En este capítulo abordaremos una breve introducción a los modelos de aprendizaje automático, en concreto los modelos LLM (*Large Language Models*), sentando las bases para un análisis más detallado del modelo específico elegido para desarrollar este estudio, los modelos BERT. A continuación, ofreceremos una visión actual de las redes sociales y la salud mental para posteriormente pasar a examinar su impacto. Finalmente se exponen datos reales, tanto en un contexto global como nacional, para poder ver una comparativa de la magnitud del problema al que nos enfrentamos.

2.2 Introducción a Modelos LLM y Modelos BERT

Los modelos LLM representan un avance significativo en el procesamiento del lenguaje natural (PLN) y la IA. Utilizan redes neuronales de millones de parámetros que son entrenadas en conjuntos de datos de texto sin etiquetar mediante aprendizaje supervisado [4].

En inicio, la introducción de las redes neuronales recurrentes (RNR) fue crucial para para confeccionar el modelado secuencial de datos, pero se enfrentaban a grandes problemas como los gradientes de fuga o las dependencias a largo plazo. Para solventarlo, se implementó una arquitectura de transformadores con un mecanismo de autoatención la cual constituiría la base para futuros modelos LLM avanzados como las variantes de los modelos BERT o GPT de OpenAI [4, 5].

Actualmente, la arquitectura de transformadores se encuentra presente en modelos como el codificador bidireccional de Transformadores (BERT) y los Transformadores unidireccionales Generativos Pre-trained Transformers (GPT) para superar las limitaciones de modelos anteriores, por ejemplo los modelos basados en RNRs [6].

Google abordó las limitaciones de la comprensión contextual mediante utilizando representaciones bidireccionales profundas en todas las capas [7, 8]. Para ello utiliza una arquitectura de transformers para generar representaciones contextuales del texto, lo que le permite captar con mayor precisión el significado de las palabras en su contexto.

Por todo lo anterior, los modelos BERT fueron el modelo elegido para desarrollar el presente trabajo debido a su eficacia en diversas aplicaciones de PLN. Centrándonos en el análisis de sentimientos, este tipo de modelos son capaces de interpretar las emociones subyacentes en el texto y predecir con exactitud el tono emocional [9].

2.3 Estado del arte en redes sociales y salud mental

2.3.1 Impacto de las redes sociales

Según la OMS (Organización Mundial de la Salud), los trastornos de salud mental son una de las principales causas de enfermedad y discapacidad en los adolescentes. Investigaciones recientes han mostrado que la pandemia y el confinamiento por la COVID-19 han contribuido al reciente aumento de estos trastornos [10]. Aunque la ansiedad y la depresión son los trastornos más frecuentes en adolescentes, se ha detectado un incremento en otros problemas psicológicos, como pensamientos suicidas, baja autoestima, trastornos de la alimentación, autolesiones y agresividad [11]. Según datos de Unicef, España es el país líder en Europa en prevalencia de problemas de salud mental entre niños y adolescentes [12]. Siguiendo esta línea, estudios internacionales afirman que España es el país donde el uso de las redes sociales por parte de los adolescentes es más problemático.

En 2024, se estima que alrededor de 5.000 millones de personas utilizan redes sociales mensualmente, con una mayoría procedente de Europa, Asia y América. Esta amplia adopción en dichas regiones se refleja no solo en la cantidad de usuarios, sino también en el tiempo que dedican a estas plataformas. A comienzos de 2025, los países del norte y oeste de Europa lideraban el ranking global de penetración en redes sociales, superando ambos el 77% de la población conectada [13].

Las estadísticas muestran que TikTok es la red social más popular entre los jóvenes a nivel mundial, superando con creces a otras plataformas. En España, es la preferida por los menores y la que más utilizan, con un consumo promedio de alrededor de una hora y

media al día. Estos datos colocan a España como el país con el mayor porcentaje de adolescentes usuarios de TikTok.

Respecto a las estadísticas de Instagram, en 2024, más de un tercio de la población mundial de Instagram en todo el mundo tiene 34 años o menos y el 16,4% de los usuarios activos eran hombres de entre 18 y 24 años. De hecho, Instagram tuvo un ascenso interanual en el tamaño de su audiencia superior al 5%, valor gracias al cual se convierte en la red social que más incrementó su número de usuarios. En España a fecha en 2024, Instagram sigue siendo una herramienta que goza de mayor penetración entre los internautas de más de 34 años. Así lo demuestra el hecho de que más de tres quintas partes de los usuarios tengan edades comprendidas entre los 35 y los 74 años [13,14].

2.3.2 Visión de las redes sociales y la salud mental en el contexto actual

El uso de medios digitales ha ido crecido rápidamente en los últimos años, al igual que los problemas de salud mental en niños y adolescentes. Este hecho ha generado un debate sobre el papel que juegan los medios digitales como factor de riesgo y desencadenante de los problemas de salud mental [15].

Diversos artículos han identificado efectos a largo plazo entre un elevado tiempo de exposición a pantallas y la mala calidad del sueño y/o síntomas depresivos. El uso diario de pantallas superior al rango de 2 a 4 horas se ha asociado con resultados adversos para la salud mental incluyendo sobrepeso y obesidad, dolor de espalda, dolor de cabeza y síntomas de depresión y ansiedad, siendo estos últimos especialmente atribuidos al creciente uso de las redes sociales [16, 17].

Las redes sociales ofrecen a los adolescentes nuevas formas de medir la aprobación social de su entorno, como la cantidad de likes o el número de comentarios en sus publicaciones. Para algunos, este seguimiento constante de las reacciones e interacciones de sus pares puede provocar un aumento de la ansiedad, minar su autoestima o intensificar el impacto de las opiniones ajenas en su persona [18].

En los casos más extremos pueden llegar incluso afectar en el desarrollo de identidad personal de los adolescentes y llevar a ideas suicidas, provocando además la creación de hábitos tóxicos recurrentes en relación con el uso de las redes sociales [19].

Según la Organización Mundial de la Salud (OMS), alrededor de uno de cada siete adolescentes de entre 10 y 19 años sufre algún trastorno mental, lo que equivale al 14% de la población juvenil a nivel mundial.

Los trastornos más comunes en este grupo de edad son la depresión y la ansiedad, que afectan al 5,5% de los adolescentes de 15 a 19 años. Asimismo, la OMS señala que el suicidio constituye la tercera causa de defunción entre las personas de 15 a 29 años, evidenciando la gravedad de los trastornos de salud mental en esta etapa de la vida [20].

Según el Informe del Departamento de Salud y Servicios Humanos de EE. UU, el 66% de los adolescentes utilizan redes sociales todos los días y un 33% lo hace de manera constante.

Los efectos más comunes asociados al uso excesivo de las redes sociales incluyen aumento de la ansiedad, depresión, problemas de sueño y baja autoestima. Un 46% de los adolescentes considera que las redes sociales dañan su autopercepción corporal y un 70% confiesa dificultades para dormir cuando tienen un uso frecuente de las redes sociales antes de acostarse.

El informe apunta que los adolescentes que pasan más de 3 horas en redes sociales tienen el doble de probabilidades de experimentar problemas de salud mental destacando la ansiedad y la baja autoestima. Además, las redes sociales contribuyen a una mayor comparación social, lo que puede hacer que los jóvenes se sientan más inseguros [21].

Siguiendo con la comparativa, el informe sobre el impacto del aumento del uso de Internet y las redes sociales del Observatorio Nacional de Tecnología y Sociedad (ONTSI) en España, un 11,3% de las personas entre 15 y 24 años se encuentran en riesgo elevado de desarrollar un uso compulsivo de servicios digitales. Este porcentaje aumenta al 33% en el grupo de 12 a 16 años. El mismo informe señala que el 44,6% de los estudiantes siente que el uso de redes sociales les resta tiempo de estudio. Además, un 12,9% ha reducido su actividad deportiva.

El uso elevado de redes sociales se asocia con un aumento de la soledad y una menor interacción social cara a cara, lo que impacta en la salud mental de los jóvenes. Un 9,4% pasa menos tiempo con amistades, y un 26% se siente más solo desde que utiliza dispositivos tecnológicos [22].

En este entorno de preocupación social, la Inteligencia Artificial (IA), especialmente el Procesamiento del Lenguaje Natural (PLN) y los modelos de Representaciones Codificadoras Bidireccionales (BERT), han revolucionado el análisis de datos al permitir una comprensión más profunda y precisa de grandes volúmenes de texto. Estas tecnologías son cruciales para analizar las respuestas emocionales (niveles de polaridad, emociones), detectando matices y patrones que antes eran difíciles de identificar. El análisis de sentimientos es, por tanto, una subcategoría de la PNL, que otorga a los ordenadores la capacidad de comprender el lenguaje humano escrito o hablado. Su integración en el campo de la salud mental es pionera, sobre todo en las aplicaciones de redes sociales, donde puede aportar información valiosa sobre el bienestar emocional de los usuarios y la detección de patrones de comportamiento.

3

Herramientas utilizadas

3.1 Introducción

Para realizar el presente estudio se han utilizado las siguientes herramientas y plataformas software que se describen a continuación.

3.2 Python

Python es un lenguaje de programación de alto nivel creado en los años 90 por Guido van Rossum. Se trata de un lenguaje interpretado, orientado a objetos, multiplataforma y con tipado dinámico. Para ponernos en contexto, un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete o Shell. El intérprete ejecuta el código en vez de directamente compilar el código a lenguaje máquina y ejecutarlo en un ordenador (lo que conocemos como lenguaje interpretado).

Se ha elegido el lenguaje Python para este trabajo ya que presenta una sintaxis muy limpia lo que favorece la lectura del código. Además, es uno de los lenguajes más utilizados en la actualidad en las investigaciones debido a que cuenta con extensas bibliotecas dotadas de herramientas y algoritmos especializados en IA [23].

Para llevar a cabo la implementación y evaluación del modelo BERT, emplearemos diversas librerías que facilitarán el manejo de datos, el proceso de entrenamiento, la evaluación del modelo y la visualización de los resultados obtenidos.

En particular, utilizaremos las siguientes:

- **Pandas**: Librería diseñada para facilitar el manejo eficiente de datos. Ofrece estructuras especializadas, como los dataframes, que permiten organizar y procesar la información de manera intuitiva y flexible [24].

- **Numpy**: Librería fundamental para el cálculo numérico en Python, que ofrece herramientas optimizadas para manipular matrices y *arrays* de datos multidimensionales de manera eficiente [25].
- **Matplotlib.pyplot**: Librería de visualización en Python que facilita la creación de gráficos y representaciones de datos de forma sencilla [26].
- **Hugging Face Transformers**: Librería desarrollada por la comunidad de Hugging Face, enfocada en la experimentación con modelos basados en transformers para tareas como procesamiento de lenguaje natural, entre otras. Proporciona documentación detallada para utilizar, entrenar y compartir modelos [27].

3.3 Google Colab

Google Colaboratory, conocido como Google Colab, es una plataforma de desarrollo en la nube diseñada por Google que facilita la colaboración de cuadernos interactivos basados en Jupyter.

Estos cuadernos se ejecutan en los servidores de Google, brindando acceso a recursos computacionales sin costo, pero limitados para la ejecución de código. Además, Google Colab cuenta con una interfaz sencilla y herramientas preinstaladas que permiten llevar a cabo tareas de aprendizaje automático y análisis de datos de manera eficaz [28].

La ventaja por la cual se ha seleccionado para realizar el trabajo es su facilidad de uso y conectividad ya que se encuentra en la nube y permite el acceso a GPUs, un potente recurso para llevar a cabo preprocesamiento de los datos para el posterior entrenamiento de los modelos de aprendizaje automático.

3.4 Anaconda

Anaconda es un software gratuito que ofrece un conjunto de herramientas orientadas a la investigación. Su instalación permite acceder a distintos entornos para programar en Python o R. Estos entornos, conocidos como entornos de desarrollo integrado (IDE), proporcionan diversas funcionalidades que facilitan la escritura, edición y depuración de código, además de permitir la visualización de datos, gestión de variables, presentación de resultados y colaboración en proyectos.

Al descargar el kit de herramientas, se obtiene acceso a una amplia variedad de funciones predefinidas desarrolladas por la comunidad de Python. Estas funciones se organizan en bibliotecas, las cuales pueden instalarse de manera sencilla a través de Anaconda [29].

3.5 Jupyter Notebook

Jupyter Notebook es una innovadora plataforma web diseñada para desarrollar y compartir documentos computacionales. Ofrece una experiencia de usuario intuitiva y eficiente, centrada en la creación y administración de documentos interactivos.

Su mayor ventaja en este proyecto radica en su entorno interactivo, que permite ajustar parámetros y visualizar resultados en tiempo real. Al combinar código, gráficos y anotaciones en un solo documento, facilita tanto la optimización y mejora del modelo como una mejor comprensión y documentación del proceso [30].

3.6 Wandb

Wandb es una plataforma especializada en seguimientos para la gestión de la optimización de los modelos de aprendizaje automático [31]. A través de su API, se ha utilizado para la búsqueda de hiperparámetros y métricas del sistema de los modelos que se van a optimizar.

3.7 Instagram

Instagram es una red social enfocada en compartir fotos, videos y contenido interactivo y publicitario. Fue lanzada en 2010 y desde entonces ha ido ganando popularidad hasta consolidarse como una de las principales redes sociales. Actualmente es propiedad de Meta. La plataforma permite a los usuarios conectarse con otros usuarios a través de *likes*, comentarios, mensajes directos o respondiendo a *stories* (fotos o videos que se eliminan en el transcurso de 24 horas) [32].

Para el presente estudio se han recopilado comentarios en diferentes publicaciones sobre salud mental para analizar su respuesta emocional y encontrar posibles patrones.

3.8 Métricas de rendimiento

Antes de finalizar el estudio de las herramientas utilizadas, es muy importante conocer cómo se evalúa la eficacia del modelo de clasificación. Para ello vamos a presentar diferentes métricas comúnmente utilizadas en estudios de este tipo para así, poder realizar comparativas con estudios anteriores.

Considerando un modelo de clasificación de emociones destinado a determinar si un comentario expresa presencia o ausencia de gratitud. Al realizar una predicción, se pueden obtener 4 posibles resultados en función de la relación entre la salida esperada y el resultado finalmente obtenido.

- ***True positive* (TP)**: Verdadero positivo, este clasifica correctamente la presencia de gratitud en un comentario que lo contiene.
- ***False positive* (FP)**: Falso positivo, este clasifica erróneamente la presencia de gratitud en un comentario que no lo contiene.
- ***True negative* (TN)**: Verdadero negativo, este clasifica correctamente la ausencia de gratitud en un comentario que no lo contiene.
- ***False negative* (FN)**: Falso negativo, este clasifica erróneamente la ausencia de gratitud en un comentario que si lo contiene.

Para hacernos una idea más clara en la Figura 1 representamos una matriz de confusión. En las filas se representan las etiquetas verdaderas (presencia o ausencia de gratitud) y en las columnas se representan las predicciones del clasificador.

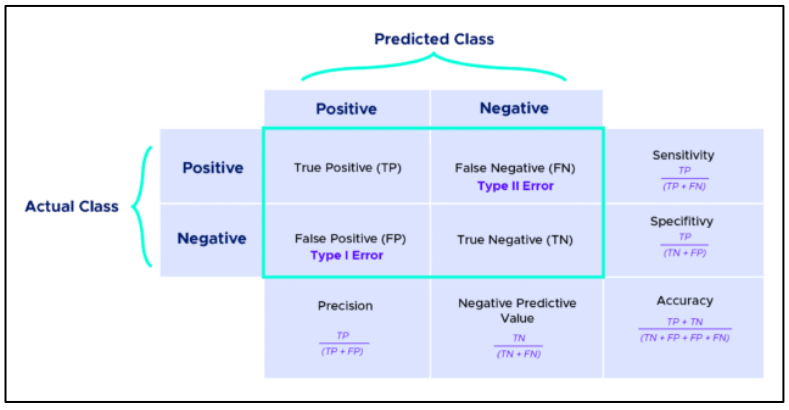


Figura 1: Matriz de confusión [33].

Además, para poder evaluar el rendimiento de los modelos de clasificación, se han utilizado diversas métricas basadas en los resultados de la matriz de confusión que también podemos ver en la Figura 1.

- *Accuracy* (Exactitud): Métrica representada que mide la proporción de clasificaciones correctas sobre el total de predicciones.
- *Precision* (Precisión): Métrica que mide la proporción de verdaderos positivos entre todos los casos que fueron clasificados como positivos. Así sabemos la capacidad del modelo para evitar falsos positivos.
- *Recall* (Exhaustividad): Métrica que mide la proporción de verdaderos positivos entre los casos realmente positivos. Así el modelo reflejando la capacidad para evitar falsos negativos. En la Figura 1 la encontramos bajo el nombre de *Sensitivity*.
- *F1-score* (Puntuación F1): Métrica obtenida de la media entre la *Precision* y el *Recall*. Nos es útil para evaluar el rendimiento de modelos en escenarios donde las clases no están igualmente representadas.

4

Ampliación del Corpus de Salud Mental en Instagram

4.1 Introducción

Este capítulo se enfocará en la descripción del corpus inicial y su posterior ampliación mediante la incorporación de nuevos comentarios extraídos de publicaciones en redes sociales, en concreto en Instagram. El propósito principal es mejorar la precisión y solidez de los modelos de clasificación empleados para lograr una clasificación más efectiva de los comentarios que dichas publicaciones reciben según su polaridad (Positiva, Indeterminada/Neutral o Negativa) y/o según la emoción que expresan (Amor/Admiración, Comprensión/Empatía/Identificación, Enfado/Desprecio/Burla, Gratitud, Tristeza/Pena o Indeterminada/Neutral). Para alcanzar este objetivo, se ha implementado una metodología específica para la selección tanto de publicaciones como de *Influencers*, cuyos detalles se expondrán a lo largo de este capítulo.

4.2 Descripción del corpus inicial y sus categorías

El corpus o base de datos inicial es un fichero que recoge los mensajes escritos en comentarios de los diferentes *posts* utilizados. En primera instancia el fichero contenía 2287 comentarios obtenidos de publicaciones en Instagram relacionados con la salud mental. A continuación, en este TFG se añadieron 2086 comentarios siguiendo la misma temática que los iniciales, esto es, centrándonos en la búsqueda e integración de *posts* realizados por mujeres.

En el corpus final podemos encontrar 4373 comentarios etiquetados con polaridad, emociones y estigma además del nombre de la *Influencer* asociado a cada comentario. A continuación, se va a describir con detenimiento cada una de las categorías del corpus.

1. **Polaridad**: En esta columna del corpus encontraremos una variable que representa el sentimiento de polaridad del comentario medido en tres niveles:

    - Positiva: Este nivel es utilizado para aquellos comentarios que reflejen un sentimiento de amor, comprensión o admiración.
    - Indeterminada/Neutral: En este nivel se encuentran los comentarios que presentan una polaridad ambigua, es decir, los que no presentan un sentimiento claro.
    - Negativa: El contrapunto del primer nivel mencionado, es utilizado cuando se expresan sentimientos despectivos como ira, burla o sarcasmo.

2. **Estigma**: Al trabajar sobre un tema controversial como la salud mental, hemos encontrado comentarios estigmatizantes, indicados en esta columna con un "Sí" o un "No". Este estigma es expresado de varias formas, desde el rechazo o pena pasando por el desprecio y la burla e incluso llegando a la ira hacia la persona. También debemos tener en cuenta los comentarios en tono irónico o sarcástico y las alusiones a la comercialización y mercadeo de la salud mental.

3. **Emociones**: En esta columna del corpus encontramos otra categoría relevante de este estudio asociada a la emoción del comentario. Para ello, hemos utilizado un conjunto de emociones definidas con las etiquetas que presentamos a continuación presentes en los modelos de Plutchick (2001), (Ekman, 2004) y Fredrickson (2013):

    - Amor/Admiración: Emoción positiva donde el apoyo, la admiración, la confianza y el amor guardan una estrecha relación entre sí. Los comentarios presentan todo cálido y elogioso.
    - Comprensión/Empatía/Identificación: Emoción positiva entendida como la comprensión del contexto del mensaje y la proyección de uno mismo en la misma situación. En los comentarios se muestra el desarrollo de la empatía y comprensión hacia la otra persona debido a vivencias similares.
    - Enfado/Desprecio/Burla: Emoción negativa caracterizada por presentar enojo, irritación o furia. Los comentarios muestran rechazo o desprecio hacia el contenido mostrado debido a una percepción de menosprecio y ridiculización de la salud mental.

- Gratitud: Emoción positiva proveniente del impulso de amabilidad y generosidad hacia la otra persona. Los comentarios que denotan esta emoción están caracterizados por el agradecimiento.
- Tristeza/Pena: Emoción negativa que se presenta ante situaciones que implican lástima hacia la otra persona o en eventos desagradables. Los comentarios muestran desconsuelo y aflicción por el mensaje recibido.
- Indeterminada/Neutral: Además, añadimos esta categoría para aquellos comentarios que presenten una emoción ambigua o que nos sea imposible de categorizar, aquellos que presenten una connotación religiosa o mensajes que no presentan ninguna emoción.

4.3 Metodología para ampliar el Corpus

4.3.1 Criterios de selección de nuevos posts

El proceso de identificación y obtención de publicaciones es un aspecto crucial en el estudio de las reacciones emocionales y el análisis de las redes sociales. Se llevó a cabo una búsqueda exhaustiva en Instagram para encontrar publicaciones donde las autoras abordaran temas de salud mental desde una perspectiva íntima, compartiendo sus vivencias y dificultades personales. Con el fin de garantizar el impacto de estas publicaciones, se estableció como criterio de selección que fueran *Influencers* españolas con un número significativo de seguidores en Instagram. La metodología empleada fue la siguiente:

1. **Identificación de posts**: Se realizó una búsqueda de publicaciones que abordaran cuestiones de salud mental desde una perspectiva personal. El enfoque se centró en experiencias relacionadas con trastornos como la depresión, la ansiedad y otros problemas psicológicos. Para optimizar la búsqueda, se emplearon términos clave y etiquetas específicas vinculadas a la salud mental.
2. **Criterios de alcance**: Partiendo de las publicaciones seleccionadas en el punto anterior, solo se tuvieron en cuenta aquellas cuyas autoras tuvieran un número mínimo de seguidores, estableciendo ese umbral en 10.000. Siguiendo esta estrategia nos aseguramos de obtener una gran variedad de comentarios debidos a los altos niveles de interacción con la publicación.

3. **Recopilación de datos**: Finalmente, de los posts restantes que cumplían ambas condiciones anteriores se hizo una selección manual de los comentarios de cada publicación para llevarlos a un documento excel para su posterior etiquetado.

Es fundamental implementar esta metodología para lograr los resultados óptimos y poder obtener un modelo fiable.

4.3.2 Descripción de los posts seleccionados

En esta sección se abordarán las publicaciones recopiladas y el procedimiento de etiquetado que se aplicará a ellas, las cuales se utilizarán para modelar y entrenar el modelo de clasificación. Estas publicaciones fueron seleccionadas siguiendo la metodología detallada en la sección anterior.

Todos los posts seleccionados son de mujeres españolas influyentes en Instagram. La mayoría consiste en relatos personales sobre salud mental, destinados a aumentar la visibilidad del tema, las cuales suelen recibir comentarios positivos y son bien aceptadas por sus seguidores. Sin embargo, algunas de las publicaciones seleccionadas representan situaciones ficticias que ilustran posibles problemas de salud mental que podrían padecer las *Influencers*, que fueron recibidas negativamente y han generado numerosas críticas. La variedad de publicaciones aborda el mismo tema, pero desde distintas perspectivas para poder crear un corpus más diverso y así realizar un mejor entrenamiento del modelo.

A continuación, se presenta la Tabla 1, que resume los datos clave de los posts descargados y sus protagonistas. Posteriormente, se procederá a explicar de manera detallada cada uno de los posts utilizados para el entrenamiento del modelo, así como las protagonistas de estos.

Influencer	Followers	Instagram Post	Number of responses	Number of selected comments
Angie	356 mil	https://www.instagram.com/p/CyN0fDhM5sw/?igshid=MTc4MmM1YmI2Ng==	508	49
Dafne Fernández	306 mil	https://www.instagram.com/p/Cx-bE-ishAB/?igshid=MTc4MmM1YmI2Ng==	96	72
Chloe Wallace	104 mil	https://www.instagram.com/p/CyN-ynBI4J3/?igshid=MTc4MmM1YmI2Ng==	199	78

Tamara Gorro	2 millones	https://www.instagram.com/p/CbKt5p-lSb2/?utm_source=ig_web_copy_link	868	59
Marta Riesco	191 mil	https://www.instagram.com/p/CcP1m6Sjec5/?utm_source=ig_web_copy_link	798	80
Gloria Camila Ortega	858 mil	https://www.instagram.com/reel/CqaMD3qraDs/?igshid=emNjZ2xxbGFhdTYx	6298	169

Tabla 1. Post y número de comentarios descargados.

- **Angie (@angynas):**

  Cantante, actriz, creadora de contenido y recientemente escritora. En Instagram cuenta con 3006 publicaciones y más de 356 mil seguidores. En él podemos ver promociones de sus trabajos, viajes y sobre todo un espacio donde comparte mensajes enfocados en el bienestar emocional y la salud mental. En el post seleccionado aparece un carrusel de imágenes. En la primera de ellas aparece la actriz sonriendo y el resto son imágenes con mensajes como: *"CELÉBRATE, PORQUE HAS HECHO SACRIFICIOS QUE MUCHOS NO ENTIENDEN"* o *"¡SOY UN SER EMOCIONAL! Y ESTÁ BIEN SERLO"*, las cuales acompaña con el siguiente texto:

  "HOY me celebro porque lo estoy haciendo bien porque he pedido ayuda, porque estoy mejorando, porque he empezado a hacer ejercicio, porque se de donde viene mi tristeza, porque soy valiente. No me avergüenzo de mis emociones, ni de mi proceso. He sentido muchas veces que no lo hacía bien, casi a diario. Hoy tú puedes sentir que no estás avanzando. Se más compasiva contigo. Lo estás haciendo como puedes ahora, con las herramientas que tienes. Lo importante es seguir intentándolo. No te rindas, porfa. No estás sola. No estás solo. Ten paciencia contigo. Deja de machacarte. Todos cometemos errores. Todos nos caemos. Todos empezamos de cero. Todos lloramos y sufrimos. No te culpabilices. Si no tienes ganas de sonreír, no lo hagas. Si tienes ganas de llorar, LLORA. No es malo. Libérate de tanta carga y si no puedes tú sola, pide ayuda. Familiares, amigos y profesionales. Alguien te escucha y te quiere. Quiérete, eres un ser excepcional. Te quiero persona bonita que me estás leyendo. Un día a la vez. Si hoy no puedes, mañana podrás. #diamundialdelasaludmental. "

Podemos apreciar como hace una reflexión personal sobre su experiencia en el ámbito de la salud mental, destacando la importancia de cuidar de uno mismo y a reconocer que cada paso es parte del proceso de sanación y crecimiento personal. También lanza un mensaje de apoyo y comprensión hacia sus seguidores que puedan estar pasando por una situación similar. Entre los comentarios recibidos destacamos los mensajes de amor y sobre todo la identificación de sus seguidores aportando sus experiencias personales.

- **Dafne Fernández (@dafnefernandez):**
  Actriz, modelo y creadora de contenido. En Instagram podemos ver como combina su vida profesional con momentos personales siempre con un mensaje positivo que siguen más de 306 mil personas. El post elegido es acompañado por el siguiente texto:

  "He estado un par de días en un balneario. Sola. Quería cuidar mi salud mental. Apartarme un poco del ruido y estar conmigo misma en solitud. Ahora estoy más preparada para afrontar el rodaje de la segunda temporada de #4estrellas y estar menos irascible en general. Esta práctica debería ser obligatoria dos veces al año. Ser madre y trabajadora es más fácil cuando te cuidas. Elegí @castillatermalbrihuega después de leer todas vuestras recomendaciones y es justo lo que buscaba. Un lugar hermoso, tranquilo y cerca de mi hogar. Os dejo un reel porque compartir es de guapas 🤓"

  El post fue bien recibido por sus seguidores donde los comentarios principalmente se centraban en muestras de identificación, recomendaciones futuras y mensajes de cariño hacia la actriz.

- **Chloe Wallace (@chloewallace_):**
  Influencer y creadora de contenido. Su Instagram se centra en mostrar su característico estilo de vida y su particular enfoque de la moda y la belleza. En el post seleccionado podemos ver un *selfie* y una foto con un texto sobre una experiencia personal además de la siguiente descripción:

  "hoy es el día de la salud mental, me he enterado por mi amiga ana, que lo ha compartido en stories. Yo llevo 4 meses tomando antidepresivos. la foto (me hace mucha gracia que en la sudadera ponga fucking awesome y yo esté llorando) es de

cuando me di cuenta de que las cosas estaban realmente mal, el texto, de justo después de hacerme la foto. pero tardé un mes en empezar a tomar fluoxetina, pastillas que ya había tomado hace doce años. sentí que retrocedía en el tiempo, pero así me sentía cada vez que me tumbaba: como hace más de una década, cuando me hundía en el sofá, en picado hacia un agujero que no tenía final. a pesar del yoga, de la terapia, de compartir con mis amigas. sentía que nada cambiaba. así que volví al psiquiatra. Estoy mejor que en mayo, de cuando es la foto. aún sensible. a veces tengo momentos en los que siento que no necesito seguir medicándome- todo está bien! estoy curada! luego mi familia me vuelve a poner en mi sitio: y si no tuviera trabajo? cómo me sentiría? o me rompen el corazón un poquito y el mundo se desmorona a niveles que no debería. y eso es un signo de que bueno, quizá estoy mejor, progresa adecuadamente, pero por ahora, los antidepresivos forman parte de mi vida hasta nuevo aviso.

comparto esto porque ayer fui a ver la peli nueva de nanni moretti, que me gustó mucho. la mujer del protagonista lleva yendo meses al psiquiatra y no se lo ha contado a nadie. dice que es privado. al salir de la película me preguntaba si quizá la salud mental dejara de ser tan privada, se normalizaría más. sería un tema de conversación en abierto, algo de lo que no acomplejarse. estamos todos tristes. ¿cómo no vamos a estarlo? quizá si la salud mental fuera más mainstream, no sería un privilegio, si no un derecho. quizá no sería la principal causa de muerte de la gente joven en españa. quizá. #diadelasaludmental"

El texto transmite una desgarradora realidad y una profunda reflexión sobre la salud mental y la lucha personal con la depresión y el tratamiento psiquiátrico. Es un poderoso testimonio sobre la salud mental, que aboga por una mayor apertura y comprensión, y destaca la necesidad de normalizar la conversación en torno a estos temas para combatir el estigma y apoyar a quienes lo necesitan.

- **Tamara Gorro (@tamara_gorro):**

Influencer, empresaria y colaboradora televisiva. En su Instagram muestra aspectos de su vida personal y profesional a sus más de 2 millones de seguidores. El post elegido es un video que combina clips de video en color y blanco y negro. En los primeros podemos verla sonriendo y riendo, mientras que en los segundos aparece seria y

apenada haciendo una referencia a las luces y sombras de las emociones. Además, lo acompaña con el siguiente texto:

"¿Sabéis una cosa que he aprendido en terapia? Cierto es, que tardé bastante en comprenderlo, porque yo misma me castigaba, pero lo conseguí entender. Desestigmatizar mi enfermedad, pero no despatologizarla.
Hay que seguir caminando, dentro de las limitaciones que supone. Por estar enferma no tengo que meterme en una cueva, es más tengo que asumir que ahora forma parte de mi vida. Trabajar, reír, disfrutar, llorar si hace falta, caerme y volverme a levantar. Porque este camino es asi, pero se que también tiene fin, o al menos así visualizo yo el futuro, aunque ahora no le vea final. Eso si, echo y mucho de menos que mi cabeza no tenga que depender de una medicación e infinidad de pastillas. Familia virtual, no hay que tener una enfermedad mental para caerse, o vivir un gran bajón. Lo puedes tener puntual claro que si.
Y eso no quiere decir que no puedas sonreír. ¿Sabes por qué?, muy sencillo: Tú lo exteriorizas cuando quieres y de la manera que te nazca, sol@ o acompañad@.

Bastante difícil es superar cualquier obstáculo, como para ponerlo más complicado nosotros mismos. #mamamolona #saludmental #superacion #💪"

En él podemos encontrar un mensaje de aceptación y resiliencia en el proceso de vivir con una enfermedad mental. Notamos como hace una reflexión sobre la importancia de reconocer la enfermedad sin dejar que ésta defina su vida por completo, trabajando la aceptación y con la esperanza de un futuro más positivo. Encontramos variedad de comentarios, la mayoría de amor e identificación con este tema, pero cabe destacar una gran cantidad de comentarios en tono de critica e ironía, incluso relacionados con el tema lucrativo sobre la frivolización por parte de la i*nfluencer* de la salud mental.

- **Marta Riesco (@marta.riesco):**
  Periodista, presentadora y creadora de contenido. Su Instagram se centra en mostrar contenido de moda, viajes y colaboraciones con marcas, a la par que publica reflexiones y aspectos personales. El post seleccionado es una foto de la protagonista con los brazos en alto sintiendo la brisa con un atardecer al fondo, acompañado de la siguiente descripción:

"💥✨Necesito evadirme de todo…de tanta crítica, de tanta exposición, de tantas provocaciones, de tantos y tantos comentarios negativos de gente que no me conoce. Necesito respirar y comprobar que sigue habiendo luz a pesar de que el túnel se vuelve más oscuro por segundos. Necesito unos días para encontrarme y para refugiarme en los míos. Soy valiente, fuerte y lo he dado todo por la razón más maravillosa y poderosa que existe, el amor. Volveré con la misma luz y con la misma alegría que me caracteriza, eso es lo prometo. Gracias por todos los que siempre estáis y por los que habéis estado…No tengo miedo, solo he perdido un poco de fuerza y muchas ganas. Me voy a recuperarlas…💥✨ (En nada estoy de vuelta♥. Elegid la salud mental por encima de todo y de todos.)"

Con esta descripción la protagonista nos transmite su necesidad de autocuidado en medio de una situación emocional complicada. Relata que necesita alejarse de las críticas, la exposición y los comentarios negativos recibidos por estar constantemente en el ojo público. La mayoría de los comentarios recibidos son una mezcla entre consejos en tono amoroso por parte de sus seguidores y duras críticas por parte de los detractores en las que aluden a su integridad como persona por no tratar estos asuntos con el respeto suficiente.

- **Gloria Camila Ortega (@gloriacamilaortega):**
  Influencer y empresaria. Es hija del famoso torero José Ortega Cano y de la fallecida cantante Rocío Jurado, lo que la ha colocado en el foco mediático desde joven. El post seleccionado es un vídeo en blanco y negro donde aparece la protagonista visiblemente afectada y llorando que acompaña con la canción *"MIENTRAS ME CURO DEL CORAZÓN"* de Karol G y el siguiente texto:

  "No quiero dar pena para nada. Es lo último que quiero, porque además, hay mucha gente que está pasando por lo mismo que yo. Estamos curándonos el corazón, el alma y la vibra. Jamás he indagado en mis temas de salud mental, más allá de lo que puedo expresar sobre mi exterior y como me siento. Pero nunca os he dado un informe médico, nunca os he dado el nombre de lo que tengo, ni la medicación que tomo, ni cuantas sesiones llevo de terapia.

Bien. Hoy he decidido que, ni puedo ni merezco el ataque gratuito denigrante que recibo día tras día, sobretodo cundo salió en la televisión la serie documental " En el nombre de Rocío ".

Hay Libertad de expresión, por supuesto, pero creo que hay límites que no debemos sobrepasar. Llevo desde julio del 2022 en terapia, con una psicóloga maravillosa, y desde diciembre 2022 en terapia con una psiquiatra.

Salí de la granja con un 90% de ansiedad.

Estuve 3 semanas sin salir de casa. Llorando día si, día también.
No vi nada ni me informé de nada de lo que había pasado en mi ausencia. Solo estuve al lado de los míos, y apoyándoles en todo. A día de hoy, me tuvieron que subir medicación por tener pesadillas heavys, por estar nerviosa diariamente, por no encontrar soluciones, y por sentirme asqueada y odiada por la sociedad en la que vivimos.

Tengo un cuadro Ansiosodepresivo. Y esto diariamente me va tirando hacia abajo. No veo luz. No quiero dar pena, repito, pero si dejar claro que soy/somos de carne y hueso, que tenemos emociones, sensibilidad y corazón. Os abanderáis de feministas, de gente con criterios y solo sois escoria. No aportáis nada a este mundo ni al de mañana. Solo aportáis más frustración, que la justicia no haga nada cuando llevo amenazas de muerte, o cuando me insultáis duramente por instagram. No sabéis que trato en mis terapias, yo no trato en especial el tema de mis haters, trato mis traumas los cuales son día sí y día también recordado por vosotros. No tenéis ni la más remota idea de mi vida, de cómo la he sentido y vivido durante estos 27 años.

Os ancláis en episodios del pasado, y no avanzáis.
Me aparté de muchas cosas, y hoy lo hago de instagram, unas semanas. Gracias a las buenas personas que todavía hay. ♡"

El texto transmite una profunda sensación de agotamiento emocional y vulnerabilidad, junto con una necesidad de expresar su experiencia sin buscar compasión, sino comprensión y respeto. También enfatiza lo difícil que ha sido enfrentar el juicio constante de los demás, especialmente desde la emisión de un documental familiar. Destacan los comentarios jocosos sobre la frivolidad del video en relación con un tema tan controversial como la salud mental. También aparecen comentarios de amor y empatía.

4.3.3 Selección y etiquetado de los comentarios de los posts

Para la selección de comentarios se han considerado varios criterios, como la diversidad de contenido, la longitud y el estilo de expresión. Se recopilaron comentarios variados con el objetivo de obtener una cantidad representativa de cada emoción y polaridad, tratando de balancear entre las clases menos representativas como "Gratitud" o "Tristeza/Pena" para asegurar una mayor precisión en los resultados del modelo. Estos comentarios fueron almacenados en un archivo en formato xlsx y etiquetados manualmente, utilizando palabras clave específicas para cada emoción.

Cabe destacar que el proceso de etiquetado manual de los comentarios fue realizado por dos expertos a ciegas (psicólogos, personas formadas) y un tercer experto revisó las discrepancias. En caso de que el tercer revisor no deshiciera el empate, estos comentarios se descartaban del corpus. Para llevar a cabo este proceso, se elaboró un conjunto de directrices sobre el etiquetado de las distintas categorías para formar a los expertos. En concreto se presentan etiquetas típicas que se eligieron para la búsqueda de comentarios de cada tipo de emoción:

- **Amor/Admiración**: te quiero, guapa, te amo, eres la mejor.
- **Comprensión/Empatía/Identificación**: Te entiendo, totalmente de acuerdo, es lo que me pasa a mí.
- **Enfado/Desprecio/Burla**: Payasa, mentira, me rio de ti.
- **Gratitud**: Gracias, bravo.
- **Tristeza/Pena**: Pobrecita, lloro, que pena.

4.4 Análisis descriptivo del corpus final

Utilizaremos el corpus de datos analizado en secciones anteriores formado por 4373 comentarios con tres campos: el texto del comentario a analizar, su polaridad y la emoción que transmite, que posteriormente se utilizarán para el entrenamiento del modelo BERT.

En la Figura 1 podemos observar la distribución de comentarios en función de su polaridad, que, como sabemos, puede ser de 3 tipos diferentes: Positiva, Negativa e Indeterminada/Neutral. Nos encontramos con la siguiente distribución 2751 positivos, 1281 negativos y 340 indeterminados o neutros.

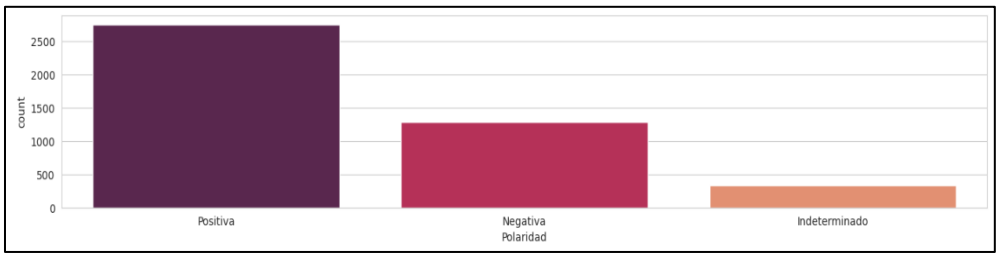


Figura 2: Distribución del corpus por polaridad.

Como podemos ver en la Figura 1, las polaridades positiva y negativa tienen una gran cantidad de muestras en comparación con la polaridad indeterminada. Este hecho nos podría llevar a un desbalance del modelo, resultando en una mayor probabilidad de predicción para las clases con más muestras. Así mismo sabemos que cada comentario solo puede pertenecer a una de las tres categorías de polaridad, por tanto, nos encontramos ante un problema de clasificación multiclase con etiqueta única.

En la Figura 2 se muestra la distribución de los comentarios en función de su emoción. En esta ocasión, la distribución de los comentarios quedaría de la siguiente manera: 1267 de Amor/Admiración, 548 de Gratitud, 293 de Tristeza/Pena, 830 de Enfado/Desprecio/Burla, 1148 de Comprensión/Empatía/Identificación y 285 de Indeterminado.

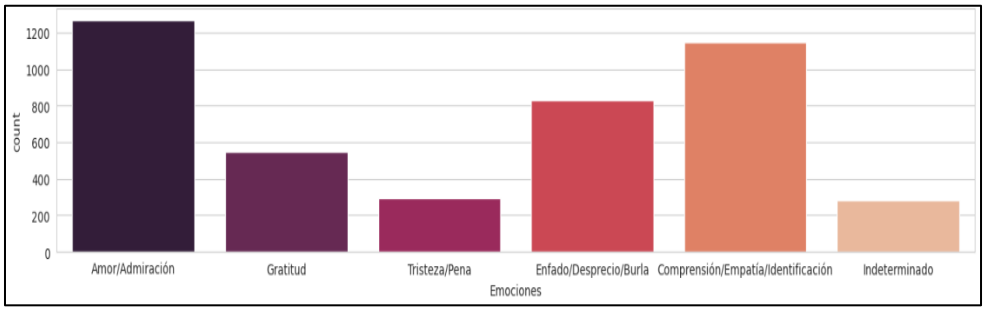


Figura 3: Distribución del corpus por emociones

Como podemos apreciar, en la Figura 2 las emociones de Amor/Admiración y Comprensión/Empatía/Identificación, seguida de la emoción de Enfado/Burla, presentan mayor cantidad de muestras en comparación con el resto de las emociones. Este hecho nos podría llevar a un desbalance del modelo, resultando en una potencial probabilidad de predicción para las clases con más muestras. Así mismo sabemos que cada comentario solo puede pertenecer a una de las seis categorías de emoción, por tanto, también nos encontramos ante un problema de clasificación multiclase con etiqueta única.

5

Análisis de Sentimientos en Instagram con modelos BERT

5.1 Introducción

Este capítulo se centrará en el análisis de la respuesta emocional de los comentarios de los posts recogidos anteriormente en Instagram empleando modelos BERT. En primer lugar, se realizará un análisis descriptivo de la base de datos inicial, para poder poner en contexto los datos obtenidos. A continuación, se empleará el modelo RoBERTuito, una variante del modelo BERT adaptada al lenguaje español y entrenada con redes sociales, que realizará la clasificación y detección emocional. Para realizar correctamente este entrenamiento, se llevará a cabo un procesamiento previo de los datos y mediante la plataforma Wandb se seleccionarán los óptimos hiperparámetros. Tras todos los pasos anteriores, se entrenará al modelo y finalmente se procederá al análisis de los resultados obtenidos para poder evaluar el rendimiento del modelo además de su capacidad y precisión para detectar polaridades y emociones extraídas de los comentarios de Instagram.

5.2 Modelo de clasificación BERT utilizado: RoBERTuito

Siguiendo con la descripción del Capítulo 2, para este estudio se empleó RoBERTuito, un modelo de lenguaje preentrenado específicamente para texto en español generado por usuarios, basado en más de 500 millones de *tweets*. Los resultados de las pruebas realizadas en un conjunto de datos de referencia mostraron que RoBERTuito superaba a otros modelos de lenguaje preentrenados en español. RoBERTuito, desarrollado en 2022, se basa en la arquitectura RoBERTa y cuenta con un tamaño de 768 dimensiones ocultas [34]. Este modelo fue entrenado por un equipo de trabajo bajo el nombre de pysentimiento [35].

La elección de este modelo de BERT se basa en su entrenamiento con comentarios de redes sociales, específicamente Twitter. Aunque el presente estudio se centra en Instagram y TikTok, la naturaleza del lenguaje empleado en estas plataformas es similar, ya que comparten dinámicas comunicativas propias del entorno digital, como la brevedad, el uso de lenguaje coloquial, emojis, abreviaciones y modismos. Además, las interacciones en redes sociales suelen reflejar tendencias lingüísticas y patrones de expresión comunes entre los usuarios, lo que permite que un modelo entrenado en una plataforma como Twitter sea aplicable a otras redes.

5.3 Preprocesamiento de los datos

Una vez terminada la ampliación del *dataset* y antes de pasar al entrenamiento del modelo debemos realizar un paso clave que es el preprocesamiento de los comentarios recogidos de las publicaciones en Instagram.

El preprocesamiento de textos es una etapa fundamental antes de entrenar modelos como BERT ya que nos garantiza una compresión homogénea de los comentarios debido a que suelen presentar faltas de ortografía y una redacción tosca. Para el preprocesado de los datos, se ha aprovechado el preprocesamiento integrado en RoBERTuito. Este proceso se ha dividido en estandarización, corrección y optimización, del siguiente modo:

- **Estandarización:** normalización de mayúsculas y minúsculas, estandarización de la risa, limitación de repeticiones de caracteres.
- **Corrección:** supresión de acentos y signos de puntuación.
- **Optimización:** entrenamiento en español, tokenización, reemplazo de hashtags por un token especial, reemplazo de emojis por su representación textual.

Este proceso de adaptación de los datos del modelo es fundamental para mejorar la precisión del análisis de sentimientos en los comentarios de Instagram. Al ajustar el modelo a las características específicas del lenguaje, se logra una mayor capacidad para detectar y clasificar las emociones expresadas en los mensajes.

Después de completar el proceso de preprocesamiento, se lleva a cabo la tokenización, que implica dividir cada palabra de los comentarios en unidades fundamentales conocidas como *tokens*. La tokenización y los *tokens* son componentes

esenciales en el procesamiento del lenguaje natural debido a que facilitan el análisis y la comprensión del significado del lenguaje por parte de los ordenadores.

5.4 Entrenamiento del modelo

El entrenamiento del modelo es la etapa crucial en la formación de un modelo para un dominio específico. En este proceso nos centraremos en la clase *Trainer*, que será la encargada de recibir el modelo y los datos específicamente preparados para poder llevar a cabo un correcto entrenamiento. Debemos notar que, al estar trabajando con datos etiquetados, necesariamente y sin excepción, los campos que contienen el comentario y el tipo de respuesta asociada se denominen *text* y *label* respectivamente, debido a consideraciones de diseño en la implantación del modelo.

5.4.1 Proceso de Tokenizado

Para entrenar un modelo con un nuevo conjunto de datos, es fundamental adaptar estos datos para que sean utilizables por el modelo. Esto requiere replicar las funciones de la capa de codificación de la arquitectura *Transformers* [36], que convierte texto en una representación numérica mediante la tokenización. La clase *Tokenizer* [37] ayuda en esta tarea proporcionando un vocabulario y métodos para transformar el texto en secuencias de *tokens*, además de agregar *tokens* especiales y aplicar técnicas de ajuste como el relleno o el truncamiento. La librería *Hugging Face* [38] ofrece un tokenizador específico adaptado a cada modelo compatible, mostrado en la Figura 4. De esta manera, nos aseguramos de que los datos se ajusten al modelo y se pueda aprovechar el mecanismo de atención de la arquitectura.

```
tokenizer = AutoTokenizer.from_pretrained("pysentimiento/robertuito-emotion-analysis")
```

Figura 4. Tokenizador de la librería *Hugging Face* para el análisis de emociones.

Para simplificar el mapeado de los datos, el conjunto se divide en datos de entrenamiento (train) y datos de prueba (*test*), convirtiendo así el *dataset* en uno compatible con la librería. Al utilizar la función del tokenización indicada en la documentación y dependiendo del tokenizador elegido, se obtienen los siguientes elementos:

- **Text**: El texto original de los comentarios.
- **Label**: La etiqueta asociada al comentario.

- **Inputs ids***:* Representación numérica del texto, el cual identifica cada palabra con un número único del vocabulario del modelo.
- **Token type ids**: Añade símbolos especiales para adaptar la entrada. En nuestro caso no se utiliza y lo representamos como una lista de ceros [39].
- **Attention masks**: Indica qué tokens deben ser considerados en el mecanismo de atención al modelo BERT [36].

La tokenización y preparación de datos son pasos fundamentales para que el modelo pueda interpretar y procesar correctamente la información del conjunto de datos, permitiendo que el entrenamiento se realice de forma eficaz.

5.4.2 Proceso de entrenamiento

Después de procesar adecuadamente los datos, comenzamos configurando los parámetros de entrenamiento con la clase *TrainingArguments*, junto con otros parámetros importantes como podemos ver en la Figura 5. En ella podemos ver los parámetros ya ajustados poder adaptar el entrenamiento del modelo a las particularidades del problema y maximizar su rendimiento.

```
training_args = TrainingArguments(report_to="wandb" ,
                                  output_dir="wandb-robertuito-emotion",
                                  overwrite_output_dir=True,
                                  evaluation_strategy="epoch",
                                  per_device_train_batch_size = 128,
                                  per_device_eval_batch_size=128, num_train_epochs = 10,
                                  fp16=True,
                                  save_strategy="epoch",
                                  load_best_model_at_end = True,
                                  metric_for_best_model="accuracy") # Necessary for earlyStop
```

Figura 5. Clase *TrainingArgs* para el análisis de emociones.

A continuación, pasamos a realizar una breve explicación de los más relevantes:

- **output_dir**: Parámetro obligatorio donde se especifica el directorio donde se guardarán los ficheros relacionados con el entrenamiento.
- **evaluation_strategy**: Especifica cuándo se llevará a cabo la evaluación del modelo utilizando el conjunto de prueba, si al final de cada época (*epoch*) o después de procesar cada lote. Podemos definir cuántas épocas o lotes deben completarse antes de cada evaluación. En este caso, la evaluación se realizará al término de una *epoch*, después de que el modelo haya procesado la totalidad de los datos de entrenamiento.

- **fp16**: Parámetro que activa el uso de punto flotante de precisión 16 bits para optimizar el uso de la memoria. Se deben usar tamaños de *batch* de hasta 128 y optimizar la velocidad de entrenamiento.
- **load_best_model_at_end** y **metric_for_best_model**: Se utilizan para implementar una parada temprana (*earlyStop*) del entrenamiento en conjunto con un *callback* en la llamada a *Trainer*. La métrica utilizada para determinar el mejor modelo (*metric_for_best_model*) es la precisión (*accuracy*).

El siguiente paso es comenzar el entrenamiento del modelo utilizando la clase *Trainer* que mostramos en la Figura 6.

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_datasets["train"],
    eval_dataset=tokenized_datasets["test"],
    compute_metrics=compute_metrics,
    tokenizer=tokenizer,
    model_init=model_init,
    data_collator=data_collator,
    callbacks=[earlyStop]
)
```

Figura 6. Objeto *Trainer*.

Al declarar un objeto *Trainer* es necesario especificar los siguientes parámetros:

- El modelo a entrenar (*model*). En la Figura 7, podemos ver el modelo utilizado para el entrenamiento para emociones.

```
model = AutoModelForSequenceClassification.from_pretrained("pysentimiento/robertuito-emotion-analysis", num_labels=6,
                                                           label2id = label2id, id2label=id2label, ignore_mismatched_sizes=True)
```

Figura 7. Modelo para el entrenamiento para emociones.

- Los parámetros de entrenamiento (*training*_args) que podemos ver en la Figura 5.
- Los conjuntos de datos de entrenamiento y prueba (*train_dataset* y *eval_dataset*).
- Las métricas de evaluación (*compute_metrics*). Estas definen en una función que devuelve un diccionario.
- El *data collator*. Es el encargado de generar los *bach* de datos para el entrenamiento, se configura con el tokenizador correspondiente.
- Algunos *callbacks* opcionales. En este caso, se utiliza un *callback earlyStop* para finalizar el entrenamiento si la última evaluación muestra una

disminución en la precisión (accuracy) en comparación con la evaluación anterior, tal y como se muestra en la Figura 8.

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1
1	No log	0.555792	0.814273	0.812869	0.814273	0.802077
2	No log	0.495276	0.844465	0.841548	0.844465	0.841723
3	No log	0.536910	0.853614	0.850876	0.853614	0.850637
4	No log	0.616935	0.861848	0.860299	0.861848	0.860418
5	0.307300	0.685880	0.860018	0.856811	0.860018	0.856265

Figura 8. Ejemplo de entrenamiento con *earlyStop*.

Para obtener un análisis detallado de las métricas por clase, se utiliza la función *evaluate* sobre el conjunto de prueba al finalizar el entrenamiento, lo que permite generar las predicciones del modelo, como se ilustra en la Figura 9.

Run summary:

eval/accuracy	0.86002
eval/f1	0.85626
eval/loss	0.68588
eval/precision	0.85681
eval/recall	0.86002
eval/runtime	5.033
eval/samples_per_second	217.168
eval/steps_per_second	27.22
train/epoch	5.0
train/global_step	515
train/learning_rate	2e-05
train/loss	0.3073

Figura 9. Análisis detallado de métricas de entrenamiento.

Al compararlo con la figura anterior, podemos ver que los resultados del desglose de métricas por clase coinciden con la quinta evaluación del modelo. Esto se debe a la parada temprana (*earlyStop*) y a los dos últimos parámetros definidos en la clase *TrainingArguments (load_best_model_at_end y metric_for_best_model*).

Para cada entrenamiento, el conjunto de datos se ha dividido utilizando la función *train_test_split*, aplicando estratificación en la variable de respuesta para asegurar que todos los posibles valores de salida estén representados en ambos subconjuntos.

5.4.3 Elección de hiperparámetros

Para conseguir el mejor rendimiento es fundamental haber seleccionado los hiperparámetros adecuados para entrenar nuestro modelo. Tenemos disponibles

infinidades de opciones para realizar esta tarea, pero nos centraremos en dos, las más populares actualmente gracias a su eficiencia y facilidad de manejo. La librería de Hugging Face [38] junto con el *backend* especializado de Wandb [40].

Para llevar a cabo esta delicada tarea, la plataforma Wandb nos simplifica el proceso ofreciéndonos una interfaz intuitiva junto con avanzadas herramientas para poder extraer los mejores resultados. A continuación, realizaremos una breve descripción de los hiperparámetros de entrenamiento sobre los cuales realizamos la búsqueda:

- ***Learning rate***: Distribución uniforme que varía entre un mínimo de 10^{-4} y un máximo de 10^{-6}. Este factor se utiliza para optimizar y acelerar la convergencia del modelo en el algoritmo de retropropagación.
- ***Per device train batch size y Per device eval batch size:*** Número de muestras utilizadas en el modelo antes del ajuste de pesos para el entrenamiento y la evaluación respectivamente. Sus valores se sitúan en potencias de 2, iniciando en 2 y acabando en 64.
- ***Num train epochs***: Distribución uniforme que varía entre un mínimo de 5 y un máximo de 15. Este factor nos indica cuando se han introducido las instancias del conjunto de entrenamiento al modelo.

A continuación, en la Figura 10 mostraremos los fragmentos de código Python que hemos utilizado para la búsqueda de hiperparámetros o *tunning* utilizando la plataforma Wandb.

Figura 10: Fragmentos de código para la conexión con Wandb, el preprocesamiento, carga de datos, preentrenamiento y búsqueda de hiperparámetros en Python.

En la Figura anterior podemos ver el detalle del proceso de inicio de sesión en la plataforma Wandb. Cuando la conexión ha sido establecida, definimos el conjunto de hiperparámetros que pasaremos a las funciones de la Figura 11.

```python
def wandb_hp_space(trial):
    return {
        "method": "random",
        "metric": {"name": "accuracy", "goal": "maximize"},
        "parameters": {
            "learning_rate": {"distribution": "uniform", "min": 1e-6, "max": 1e-4},
            "per_device_train_batch_size": {"values": [2, 4, 8, 16, 32, 64]},
            "per_device_eval_batch_size": {"values": [2, 4, 8, 16, 32, 64]},
            "num_train_epochs" : {"distribution": "int_uniform", "min": 5, "max": 15}
        },
    }

def model_init(trial):
    return AutoModelForSequenceClassification.from_pretrained(
        "pysentimiento/robertuito-emotion-analysis", ignore_mismatched_sizes=True,
        config=config, num_labels=6, label2id = label2id, id2label=id2label
    )

training_args = TrainingArguments(report_to="wandb", output_dir="wandb-robertuito-emotion", overwrite_output_dir=True, evaluation_strategy="epoch", per_device_train_batch_size = 128,
                                  per_device_eval_batch_size=128, num_train_epochs = 10, fp16=True, save_strategy="epoch", load_best_model_at_end = True,
                                  metric_for_best_model="accuracy") # Necessary for earlyStop

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_datasets["train"],
    eval_dataset=tokenized_datasets["test"],
    compute_metrics=compute_metrics,
    tokenizer=tokenizer,
    model_init=model_init,
    data_collator=data_collator,
    callbacks=[earlyStop]
)

best_trial = trainer.hyperparameter_search(
    direction="maximize",
    backend="wandb",
    hp_space=wandb_hp_space,
    n_trials=35 #pruebo con 35 y sino le bajo a 30
)


wandb.finish()
```

Figura 11: Fragmentos de código para la conexión con Wandb, selección de parámetros/valores y optimización junto con el número de entrenamientos.

En la Figura 11 se muestra un entrenamiento del modelo con los hiperparámetros definidos anteriormente. En la parte final del código se puede apreciar uno de los parámetros más importante, el parámetro *n_trials* en el que se indica el número de entrenamientos a realizar, en nuestro caso 35.

A continuación, se vamos a explicar mediante imágenes cómo navegar por la plataforma Wandb para poder visualizar y analizar las ejecuciones de las búsquedas de hiperparámetros:

1. En primer lugar, es necesario acceder a la plataforma e iniciar sesión. Para ello, se debe crear una cuenta de usuario, lo que permitirá generar una clave API. Esta clave es imprescindible para establecer la conexión entre la plataforma y el entorno de desarrollo del código.
2. La Figura 12 muestra la pantalla principal (Home) de la cuenta. En el panel lateral izquierdo se encuentran las distintas secciones disponibles. Dentro del apartado "Projects" se pueden visualizar los diferentes proyectos en los que se han almacenado los resultados de simulaciones previas. En este trabajo, se ha creado un proyecto denominado *tunning* para llevar a cabo la búsqueda de hiperparámetros.

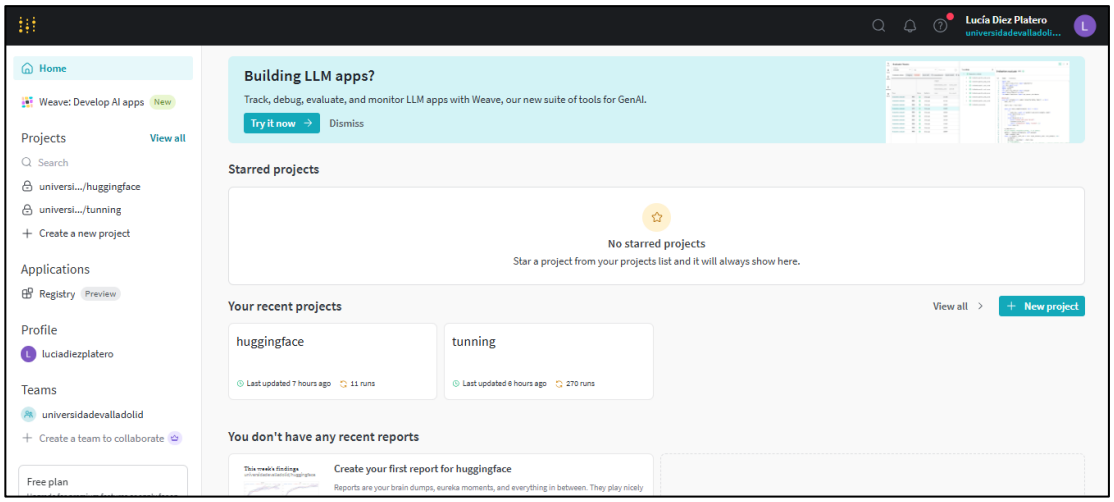


Figura 12: Repositorio *Home* en Wandb.

3. Una vez que hemos accedido al proyecto *tunning*, en el lado izquierdo de la Figura 13 seleccionaremos la opción de *Sweeps* dentro de la cual encontraremos los barridos de las simulaciones realizadas.

Figura 13: Proyecto *tunning* en Wandb.

4. En la pestaña de *Sweeps*, como podemos ver en la Figura 14, aparecen todos los barridos de las simulaciones de nuestro estudio. En este caso nos vamos a centrar en la simulación *5tb4ypn7*, la cual corresponde a la búsqueda de hiperparámetros para un análisis de emociones con 35 entrenamientos.

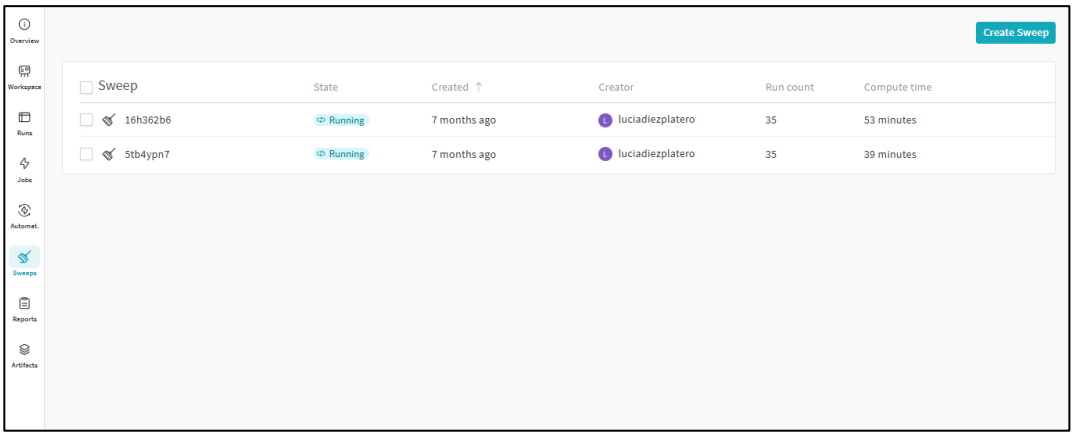


Figura 14: Pestaña *Sweeps* del proyecto *tunning* en Wandb.

5. Finalmente, en la Figura 15 vemos el resultado de la simulación *5tb4ypn7*. En ella encontramos el gráfico inferior del cual podemos obtener los valores del resultado de la búsqueda de hiperparámetros para poder entrenar el modelo.

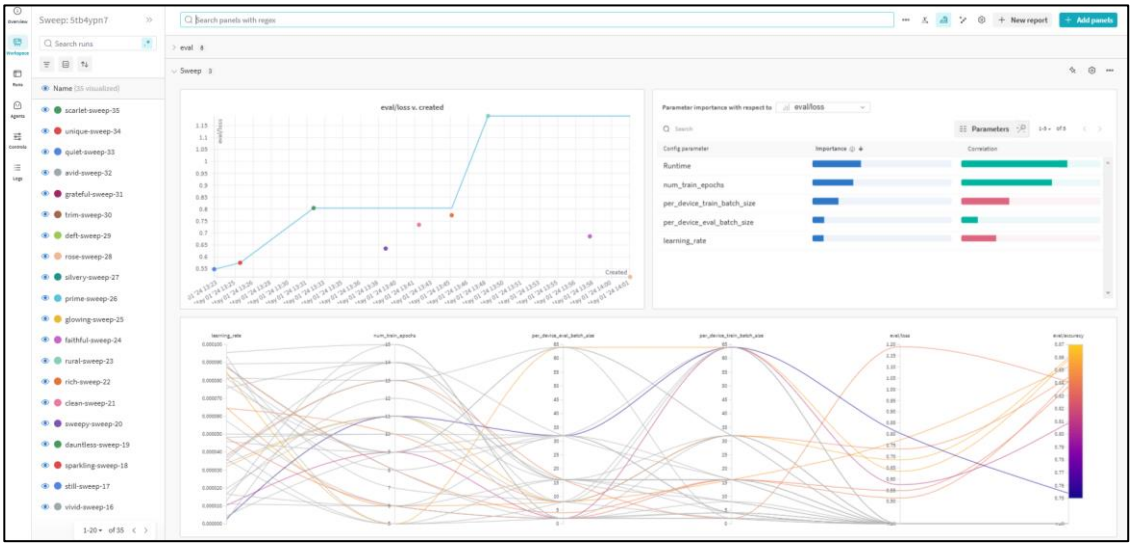


Figura 15: Búsqueda de hiperparámetros para emociones en 35 ejecuciones en Wandb.

Debemos modificar el último valor del gráfico, añadiendo la columna *eval/accurac* al lado de *eval/loss*, de esta manera, obtendremos una representación visual de los resultados en términos de precisión (*accuracy)* en lugar de pérdida (*loss)*. Para obtener el resultado de la búsqueda de hiperparámetros debemos buscar la línea más alta de la columna *eval/accuracy* y posicionar el puntero sobre ella, como podemos ver en la Figura 16.

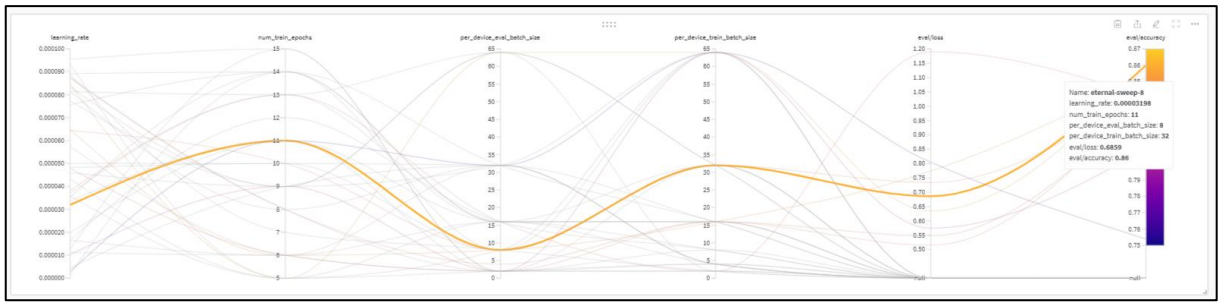


Figura 16: Visualización de los resultados de la búsqueda de hiperparámetros en Wandb.

5.5 Análisis de resultados para polaridad con RoBERTuito

Para garantizar resultados sólidos y confiables, se realizó un proceso de validación cruzada utilizando el método *10Fold* en cada ejecución con estratificación según las clases de respuesta. Este método facilitó el entrenamiento del modelo en distintas particiones del conjunto de datos y permitió calcular la media aritmética de las métricas de rendimiento para cada clase de respuesta, además de las medias generales (*macro*) y ponderadas por el tamaño de las clases (*weighted*), junto con la precisión global (*accuracy*) del modelo.

Los hiperparámetros óptimos para el entrenamiento del modelo tras su búsqueda en la plataforma Wandb [40] utilizando 35 ejecuciones son:

- learning_rate=0.00008154
- num_train_epochs=9
- per_device_train_batch_size = 64
- per_device_eval_batch_size=32

Para ordenar los resultados obtenidos en la clasificación de polaridad (Positivo, Negativo e Indeterminado) utilizando el modelo desarrollado en este trabajo, en comparación con el análisis previo realizado en el TFG [2] se presenta la Tabla 2. Cabe tener en cuenta que el análisis realizado en [2], se utilizó el corpus inicial que contenía tan solo 2288 entradas, de las cuales 1526 presentaban polaridad positiva, 587 polaridad negativa y 173 indeterminada. En nuestro caso, el corpus utilizado está formado por 4373 instancias de las cuales encontramos 2751 positivas, 1281 negativas y 340 indeterminadas. Se concluye por lo tanto que en nuestro análisis hemos utilizado para aproximadamente el doble de instancias, lo que se analizaremos en los resultados obtenidos. Los resultados se expresan mediante las métricas de precisión (Prec), sensibilidad o *recall* (Recall) y *F1-*

score (F1) para cada categoría, así como los promedios macro (*macro-avg*) y ponderado por el tamaño de las clases (*weighted-avg*), además de la precisión general (Acc). Los valores de todas las métricas se han convertido en la tabla de comparación en porcentajes para facilitar su interpretación. Tal y como se observa en la Tabla 2, el modelo RoBERTuito logra un rendimiento sólido en la clasificación de polaridad, con una precisión del 89.09%. Este nivel de precisión se ve ligeramente reducido en comparación con el alcanzado en investigaciones previas, como el trabajo fin de grado usado de referencia [2] donde se llegó al 93.3%. A pesar de ello, un análisis detallado por categorías pone de manifiesto importantes disparidades en su desempeño.

Polaridad	**Métrica**	**RoBERTuito (%) (corpus inicial)**	**RoBERTuito (%) (corpus ampliado)**
Positivo	Prec. Recall F1	95.7% 95.7% 96.5%	93.82% 93.34% 93.57%
Negativo	Prec. Recall F1	90% 91.8% 90.8%	85.88% 89.46% 87.52%
Indeterminado	Prec. Recall F1	83% 61.1% 68.9%	63.14% 53.24% 56.74%
macro	Prec. Recall F1	89.6% 83.5% 85.4%	80.95% 78.68% 79.19%
weighted	Prec. Recall F1	- - -	89.11% 89.09% 88.91%
Global	Acc	93.3%	89.09%

Tabla 2: Resultados obtenidos en clasificación de polaridad.

En resumen, la Tabla 3 ofrece una visión integral del rendimiento del modelo RoBERTuito en la clasificación de polaridad, evaluando tanto su capacidad para distinguir entre clases como su precisión global. Además, con el propósito de facilitar la comparación de la métrica de precisión (Prec.) y la precisión global (Acc.) entre el estudio anterior y el presente, se muestra la Tabla 3. Esta tabla permite analizar los cambios resultantes de las modificaciones implementadas en este trabajo y evaluar la capacidad del modelo para clasificar mensajes según su polaridad, contrastando los resultados de ambos estudios.

Polaridad	Prec. (%) (corpus inicial)	Prec. (%) (corpus final)
Positivo	95.7%	93.82%
Negativo	90%	85.88%
Indeterminado	83%	63.14%
Acc.	93.3%	89.09%

Tabla 3: Comparativa de la métrica de precisión por polaridad entre el estudio anterior [8] y el presente.

En la categoría "Positivo" el modelo destaca por su habilidad para reconocer textos con esta polaridad, con una precisión del 93.82%, un recall del 93.34% y un F1-score de 93.57%. Los resultados son muy buenos, aun situándose ligeramente por debajo de los alcanzados por el modelo anterior (Precisión: 95.7%, Recall: 95.7%, F1: 96.5%). Esta ligera reducción podría estar relacionada con el enfoque prioritario en optimizar esta clase durante el proceso de balanceo del corpus.

En la categoría "Negativo", el modelo muestra un desempeño también ligeramente inferior a la categoría anterior, alcanzando una precisión del 85.88%, un recall del 89.46% y un F1-score de 87.52%. Estos valores se siguen situando por debajo de los obtenidos por el modelo anterior (Precisión: 90%, Recall: 91.8%, F1: 90.8%). Esto sugiere que el equilibrio de clases en el corpus, orientado a incrementar los ejemplos negativos, ha influido en el rendimiento del modelo. Quizás la integración de nuevos comentarios negativos de características lingüísticas más diversas ha impactado de manera ligeramente negativa en el rendimiento del modelo para esta clase.

Finalmente, la categoría "Indeterminado/Neutral" se representa como el mayor reto para el modelo, alcanzando una precisión de 63.14%, un recall de 53.22% y un F1-score de 56.74%. Aunque estos resultados se encuentran significativamente por debajo del respecto al modelo anterior (Precisión: 83%, Recall: 61.1%, F1: 68.9%), evidencian que identificar textos sin una polaridad definida continúa siendo una tarea complicada. Esto podría explicarse por la naturaleza ambigua de este tipo de expresiones y por la escasez de muestras representativas en el corpus. Además, puede que las nuevas muestras introducidas en esta clase tengan una mayor ambigüedad o diversidad lingüística, por lo que sería necesario volver a repasar los comentarios nuevos asociados a esta clase.

En resumen, el modelo demuestra ser una herramienta efectiva para la clasificación de polaridad, destacando especialmente en la detección de textos positivos y

manteniéndose los resultados obtenidos en investigaciones anteriores [2]. Esto se debe principalmente a la ampliación y el balanceo del corpus, con un enfoque en fortalecer la representación de mensajes con polaridad negativa, lo que ha mantenido su desempeño en esta clase sin comprometer la precisión en la clase "Positivo".

No obstante, el análisis pone de manifiesto la necesidad de mejorar el rendimiento del modelo de la clase "Neutral". Para enfrentar este reto, sería posible considerar diversas estrategias, como incrementar el conjunto de datos de entrenamiento con más ejemplos de esta categoría, o quizás emplear técnicas de generación artificial de datos para ampliar la variedad de ejemplos disponibles.

El estudio de la Figura 17, conocida como matriz de confusión refuerza los hallazgos previos presentados en las Tablas 2 y 3. Los resultados evidencian un alto nivel de precisión en la detección de textos con polaridad "Positive" con 629 clasificaciones correctas y con polaridad "Negative", con 284 clasificaciones correctas.

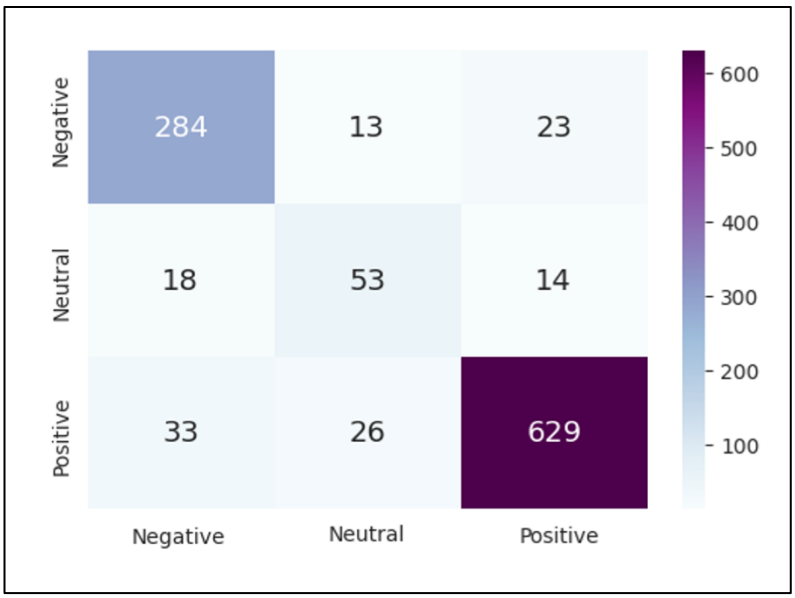


Figura 17: Matriz de confusión para la clase polaridad.

Para la clase "Neutral", el modelo presenta un rendimiento muy bajo con 53 aciertos y una distribución bastante equilibrada de errores entre las clases negativa y positiva. Esto indica que el modelo tiene dificultades para diferenciar entre textos neutrales y aquellos con polaridad, lo que se refleja en el bajo desempeño de las métricas para esta clase. Esta dificultad para clasificar textos indeterminados podría originarse en la

ambigüedad propia de esta categoría. Con frecuencia, resulta difícil determinar si un texto es verdaderamente neutral, ya que las opiniones pueden contener diversas connotaciones y suelen estar presentes simultáneamente. Además, el concepto de "neutralidad" es relativo y puede variar según el contexto en el que se interprete.

La categoría “Positive” muestra el mejor rendimiento, con 629 aciertos y una tasa de falsos positivos de 37. Sin embargo, se evidencia cierta confusión con la categoría negativa, con 33 casos erróneamente clasificados. En el caso de la categoría “Neutral”, el modelo acierta en 53 ocasiones, aunque muestra mayor confusión, con 14 textos neutrales clasificados como positivos y 18 como negativos, lo que evidencia dificultades para identificar correctamente este tipo de sentimiento. Finalmente, la categoría “Negative” muestra buen rendimiento, con 284 aciertos. Sin embargo, presenta 23 casos negativos clasificados como positivos y 13 como neutrales, quedando una tasa de falsos negativos de 36.

En conclusión, la matriz de confusión valida la capacidad del modelo para clasificar correctamente textos negativos y positivos, pero también resalta las dificultades para diferenciar textos neutrales, lo que reitera la importancia de investigar estrategias de mejora específicas para esta clase.

5.6 Análisis de resultados para emociones con RoBERTuito

Siguiendo la filosofía del aparado anterior para polaridad, los hiperparámetros óptimos para el entrenamiento del modelo tras su búsqueda en la plataforma Wandb [40] utilizando 35 ejecuciones son:

- learning_rate=0.00003198
- num_train_epochs=11
- per_device_train_batch_size=32
- per_device_eval_batch_size=8

La Tabla 4 presenta los resultados para la clasificación de las emociones. Las métricas se detallan para cada una de las clases emocionales identificadas y descritas en capítulos anteriores. Cabe tener en cuenta que el análisis realizado en el TFG [2] se utilizaron 2288 instancias, de los cuales 640 son clasificados con la emoción

Amor/Admiración, 227 de Gratitud, 122 de Tristeza/Pena, 466 de Enfado/Desprecio/Burla, 659 de Comprensión/Empatía/Identificación y 173 indeterminados, por lo que podemos concluir que hemos utilizado para nuestro análisis aproximadamente el doble de instancias, lo que se verá reflejado en los resultados obtenidos. También se presentan los promedios macro y ponderado y, se incluye la precisión global del modelo (Acc). Los valores de las métricas se han convertido a porcentajes para facilitar su interpretación. Al igual que en la sección anterior, esta tabla ofrece una visión integral del rendimiento del modelo RoBERTuito en la clasificación de emociones, evaluando tanto su capacidad para distinguir entre clases como su precisión global. El estudio de los resultados de la Tabla 4 muestra que el modelo RoBERTuito logra un rendimiento sólido en la clasificación de emociones, con una precisión del 85.58%. Este nivel de precisión es casi idéntico al alcanzado en investigaciones previas, como el trabajo fin de grado usado de referencia [2] donde se llegó al 86%, por lo que el aumento el corpus no ha impactado en gran medida en la precisión en la clasificación del modelo.

Emociones	**Métrica**	**RoBERTuito (%) (corpus inicial)**	**RoBERTuito (%) (corpus final)**
Amor/Admiración	Prec. Recall F1	93.1% 95% 94%	90.92% 92.58% 91.71%
Gratitud	Prec. Recall F1	90.8% 92.1% 91.3%	91.88% 93.79% 92.77%
Tristeza/Pena	Prec. Recall F1	83.7% 87.8% 85.1%	72.74% 75.42% 73.59%
Enfado/Desprecio/Burla	Prec. Recall F1	91% 93.2% 92%	87.63% 87.34% 87.40%
Comprensión/Empatía/Identificación	Prec. Recall F1	89.9% 88.5% 89.1%	83.59% 83.53% 83.39%
Neutral	Prec. Recall F1	79.5% 65.9% 70.9%	65% 52.43% 57.40%
macro	Prec. Recall F1	88% 87.1% 87.1%	81.96% 80.85% 81.05%
weighted	Prec. Recall F1	- - -	85.59% 85.58% 85.39%

Global	Acc.	86%	85.58%

Tabla 4: Resultados obtenidos en clasificación de emociones.

Además, con el propósito de facilitar la comparación de la métrica de precisión (Prec.) y la precisión global (Acc.) entre el estudio anterior y el presente, se incluye de modo más reducido la Tabla 5. Esta tabla permite analizar los cambios resultantes de las modificaciones implementadas en este trabajo y evaluar la capacidad del modelo para clasificar mensajes según su polaridad, contrastando los resultados de ambos estudios.

Emociones	**Prec. (%) (corpus inicial)**	**Prec. (%) (corpus final)**
Amor/Admiración	93.1%	90.92%
Gratitud	94.1%	91.88%
Tristeza/Pena	81.4%	72.74%
Enfado/Desprecio/Burla	84.7%	87.63%
Comprensión/Empatía/Identificación	84.6%	83.59%
Neutral	70.1%	65%
Acc.	86%	85.58%

Tabla 5: Comparativa de la métrica de precisión por polaridad entre el estudio anterior [2] y el presente.

Tal y como se observa en la Tabla 5, la emoción con mayor precisión es "Gratitud" con un valor de 91.88%, lo cual es notable considerando que tiene un número bajo de instancias respecto a otras emociones (548, como se detalla en la Sección 5.2). Esta alta precisión, aunque es ligeramente inferior a la de trabajos anteriores (94.1%), sugiere que los datos de entrenamiento para esta emoción eran de alta calidad y distinguibles, permitiendo al modelo aprender patrones claros para identificarla correctamente y sin una necesidad de grandes cantidades de ejemplos. En este sentido, las emociones "Amor/Admiración", "Enfado/Desprecio/Burla" y "Comprensión/Empatía/Identificación" mantienen un porcentaje de precisión similar al estudio anterior, por lo que podemos decir que se han mantenido los buenos resultados en la identificación de estas emociones. Cabe destacar que la emoción "Enfado/Desprecio/Burla" ha afinado la precisión, pasando de un 84.7% a un 87.63%.

Por el contrario, la emoción con menor precisión es "Tristeza/Pena" con un valor de 72.74%, lo cual es normal teniendo en cuenta que tiene el menor número bajo instancias (293, como se detalla en capítulos previos). Pero además se destaca su peor rendimiento respecto al análisis con el corpus inicial.

El estudio de la Figura 18, conocida como matriz de confusión refuerza los hallazgos previos presentados en las Tablas 4 y 5. La matriz de confusión muestra un buen desempeño general del modelo, especialmente en la clasificación de las emociones Love/Admiration (299/317), Comprehension/Empathy (128/137) y Anger/Mockery (183/209), con alta precisión y baja confusión. Sin embargo, hay cierta confusión entre emociones relacionadas como Gratitude, que se confunde con Love/Admiration (19 casos) y Sadness (15 casos), lo que indica una posible cercanía semántica entre estas clases. De hecho, este comportamiento puede ser debido a que, en el lenguaje, un comentario puede denotar varias emociones, por ejemplo, cuando se romantiza un problema de salud mental (denotando tristeza o pena) podemos atisbar en un mismo comentario gratitud y amor/admiración. Finalmente, las clases con menos datos como Neutral y Sadness presentan más errores relativos, sugiriendo que podrían beneficiarse de un mayor número de ejemplos o de una mejor diferenciación contextual.

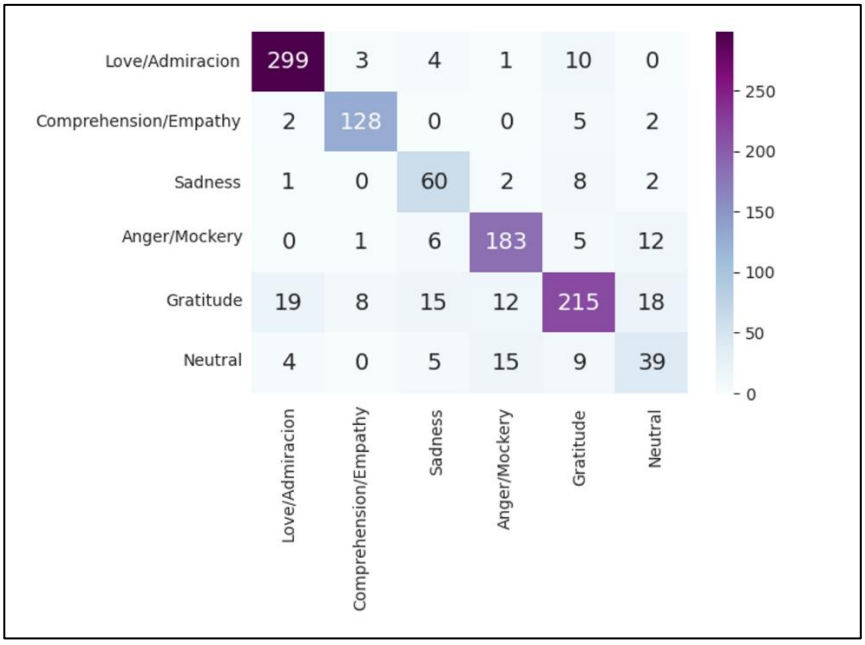


Figura 18: Matriz de confusión para la clase emociones.

Es importante señalar que la precisión global del modelo del 85.58% se mantiene similar a la de trabajos anteriores, a pesar de los ligeros descensos observadas en algunas

emociones. Esto plantea preguntas sobre si el rendimiento del modelo está limitado por la cantidad de datos de entrenamiento disponibles o si es necesario un análisis más profundo de los datos. Estas cuestiones se analizarán en detalle en las conclusiones finales y se abordarán en futuras líneas de investigación.

6

Interfaz gráfica para la implementación del modelo

6.1 Introducción

En este capítulo final nos vamos a centrar en explicar el proceso seguido para la implementación de una interfaz gráfica en la cual poder visualizar los resultados tras el entrenamiento y la posterior optimización del modelo RoBERTuito.

Esta interfaz precargamos los ficheros del modelo y del tokenizador, y posteriormente nos va a permitir tanto introducir comentarios por teclado y recibir su predicción, como la carga de ficheros con comentarios de la cual se realizará el análisis y nos mostrará tanto gráficas como una opción para poder descargar dichas predicciones.

6.2 Herramientas para la creación de la interfaz

Para su implementación se ha utilizado la herramienta Jupyter Notebook desde Anaconda utilizando como lenguaje de programación Python. Anaconda [26] es un software gratuito que permite acceder a distintos entornos de desarrollo donde poder desarrollar nuestra interfaz. Además, tiene acceso a una enorme variedad de funciones predefinidas de Python, ordenadas en bibliotecas las cuales necesitaremos más adelante. Para la instalación de Anaconda solo necesitamos acceder a su página principal y seleccionar el fichero ejecutable en función del sistema operativo en el que nos encontremos. Una vez instalado y ejecutado, en la pantalla principal encontraremos Jupyter Notebook.

6.3 Creación de la interfaz en Jupyter Notebook

Una vez hemos podido acceder a Jupyter Notebook crearemos un fichero llamado *Interfaz.ipynb* en el cual desarrollaremos nuestro código Python. A continuación, se describen los pasos para crear la interfaz en Jupyter.

6.3.1 Importación de librerías

Con el fin de implementar determinadas funcionalidades en la interfaz, fue necesario instalar las librerías correspondientes como podemos ver en la Figura 19. Para el correcto funcionamiento del código se declaran en la parte superior permitiendo así que, al ejecutar la interfaz, se carguen correctamente sus funcionalidades, asegurándonos su correcto funcionamiento.

```
import tkinter as tk
from tkinter import ttk
from tkinter import filedialog
import pandas as pd
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from transformers import AutoModelForSequenceClassification, AutoTokenizer
import torch
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from collections import Counter
from tkinter import messagebox
from IPython.display import Image
from PIL import ImageTk, Image
import os
```

Figura 19: Librerías utilizadas.

6.3.2 Creación de clases

Antes de abordar en detalle el desarrollo técnico y las definiciones del código, se considera oportuno contextualizar al lector con el objetivo de facilitar una lectura más amena y comprensible.

Las clases han sido creadas para realizar las funciones de nuestra interfaz. Así se fue asignando a cada elemento de la interfaz su correspondiente función, teniendo en cuenta que las funciones se encuentran disponibles tanto para el análisis de sentimientos como para el análisis de emociones. Antes de pasar a la explicación de las clases, se han utilizado 4 variables globales para almacenar los comentarios y los resultados tanto para

emociones como para sentimientos asegurándonos el correcto funcionamiento del código, como podemos ver en la Figura 20.

```
# Variables globales para almacenar el DataFrame
comments_df = None
results_df = None
comments_df_sentiment = None
results_df_sentiment = None
```

Figura 20: Variables globales.

Una variable global es aquella definida fuera de cualquier función y accesible desde cualquier parte del código. Se utilizan comúnmente para almacenar configuraciones, parámetros o datos que deben ser utilizados a lo largo del programa o entre diferentes funciones [41]. Se han colocado tras la definición de las librerías.

Tras las variables globales, se han creado un total de 18 clases, las cuales pasaremos a explicar a continuación según se encuentran estructuradas dentro del código.

- **def ejecutar_algoritmo(input_texts)**: En esta clase cargamos el modelo preentrenado y el tokenizador para emociones y tokenizamos junto al atributo input_texts en el cual se encuentra el texto introducido en el cuadro de texto de la interfaz. Procesamos las predicciones, las pasamos a una lista numérica de Python y mapeamos las anteriores predicciones con nuestras etiquetas, las cuales coinciden con las emociones 'Love/Admiration', 'Gratitude', 'Sadness','Anger/Mockery', 'Comprehension/Empathy' o 'Neutral'. El resultado es una de las etiquetas anteriores que aparecerá en un cuadro de texto.
- **def ejecutar_algoritmo_sentiment(input_texts_sentiment)**: Misma función que def ejecutar_algoritmo(input_texts) salvo que en este caso el modelo pre**e**ntrenado y el tokenizador es para sentimientos y las etiquetas coinciden con los sentimientos ‘Negative', 'Positive' o 'Neutral'.
- **def boton_algoritmo()**: Su propósito es invocar la función ejecutar_algoritmo(input_texts) y mostrar el resultado devuelto en un cuadro de texto, el cual se habilita para mostrar la salida de la predicción. Esta clase se ejecuta al pulsar el botón *Run Algortithm* en la pestaña *Emotions*.
- **def boton_algoritmo_sentiment()**: Misma función que boton_algoritmo() pero ahora invocando a la función

ejecutar_algoritmo_sentiment(input_texts_sentiment). Esta clase se ejecuta al pulsar el botón *Run Algorithm* en la pestaña *Polarity*.

- **def mostrar_informacion()**: Al lado del botón *Insert csv/Excel*, tenemos un botón de información el cual contiene el un mensaje para advertir al usuario que, a la hora de introducir un fichero, los comentarios deben ir en la primera columna.
- **def cargar_archivo()**: Esta clase tiene como función cargar un fichero en formato csv o Excel para su posterior uso en la predicción. Para garantizar la compatibilidad, se implementó un filtro que restringe el tipo de fichero seleccionable. Según el formato, utilizaremos read_csv o read_excel para su lectura. Si la carga se realiza correctamente, se habilita el botón *Run Algorithm with csv/Excel*, permitiendo ejecutar la predicción sobre los datos cargados. En caso de que se produjera un error durante la carga del fichero, se mostrará el siguiente mensaje: *"Error uploading file"*. En tal situación, no se habilitará el botón *Run Algorithm with csv/Excel*, impidiendo así la ejecución del algoritmo para un fichero no compatible.
- **def cargar_archivo_sentiment()**: Misma función que def cargar_archivo() salvo que en este caso es para la pestaña *Polarity*.
- **def ejecutar_excel()**: La funcionalidad es muy similar a la de la clase def ejecutar_algoritmo(input_texts), con la diferencia de que, en lugar de recibir una cadena de texto recogida de un cuadro de texto como entrada, se le proporciona una lista con todos los comentarios del fichero csv o Excel que hemos precargado. El proceso es el mismo salvo que justo antes de devolver el resultado, se invoca a las funciones mostrar_grafica1(results_df) y mostrar_grafica2(results_df), con el objetivo de visualizar en pantalla las gráficas correspondientes a los resultados de emociones obtenidos.
- **def ejecutar_excel_sentiment()**: Misma función que ejecutar_excel() utilizando ejecutar_algoritmo_sentiment(input_texts) pero invocamos a las funciones mostrar_grafica3(results_df_sentiment) y mostrar_grafica4(results_df_sentiment), con el objetivo de visualizar en pantalla las gráficas correspondientes a los resultados de polaridad obtenidos.
- **def descargar_excel()**: Su función es descargar un fichero con los resultados obtenidos de ejecutar la clase ejecutar_excel(). Al igual que en def cargar_archivo(), se da al usuario la opción de guardarlo en formato ".xlsx" o ".csv".

- **def descargar_excel_sentiment()**: Misma función que descargar_excel() pero invocamos a la función ejecutar_excel_sentiment().
- **def mostrar_mensaje()**: Esta función se encarga de mostrar un mensaje en la interfaz cuando se ha realizado la carga de un fichero correctamente tanto para la función def cargar_archivo() como def cargar_archivo_sentiment().
- **def mostrar_gráfica1(results_df)**: Su función es generar gráficas de barras que representan, en porcentaje, la distribución de comentarios asociados a cada emoción. Para ello, recibe el atributo results_df desde la función def ejecutar_excel(), y utiliza la biblioteca matplotlib.pyplot para mostrar la gráfica.
- **def mostrar_gráfica2(results_df)**: Su función es generar una gráfica evolutiva del tipo de emoción que tiene cada comentario.
- **def mostrar_gráfica3(results_df_sentiment)**: Misma función que def mostrar_gráfica1(results_df) pero recibe el atributo results_df desde la función def ejecutar_excel_sentiment().
- **def mostrar_gráfica4(results_df_sentiement)**: Misma función que def mostrar_gráfica2(results_df) pero recibe generamos una gráfica evolutiva del tipo de polaridad.
- **def draw_separator (canvas)**: Tiene como objetivo dibujar una línea continua en la interfaz, con el fin de separar visualmente distintos elementos. Recibe como parámetro el atributo canvas, que indica el *Frame* específico en el que se debe renderizar dicha línea.

6.3.3 Creación de la interfaz visual

Una vez tenemos nuestras funciones con todos sus elementos preparados, pasamos a la creación de la ventana y sus elementos gráficos.

El desarrollo de la interfaz gráfica se ha llevado a cabo utilizando el paquete tkinter, que actúa como la interfaz de Python para el conjunto de herramientas GUI Tk. Este paquete se encuentra disponible en la mayoría de las plataformas Unix, así como en los sistemas operativos Windows [42].

Una de las consideraciones de diseño fue la de fijar el tamaño de la ventana, es decir, deshabilitando la opción de redimensionarla. Esta elección nos permite evitar problemas de visualización, como el desajuste de botones, textos o gráficos. En la Figura 21 podemos ver el código que se ha utilizado para la implementación de la ventana.

```
# Crear una instancia de la ventana principal
ventana = tk.Tk()
# Obtener las dimensiones de la pantalla
screen_width = ventana.winfo_screenwidth()
screen_height = ventana.winfo_screenheight()
ventana.title("MenTAI Interface")
ventana.configure(background='#7c1324')
resultado_strvar = tk.StringVar()
resultado_sentiment_strvar = tk.StringVar()
#para que no se pueda cambiar las dimensiones de la ventana
#hay que ponerlo delante del geometry()
ventana.resizable(width=False, height=False)
# Establecer las dimensiones de la ventana
ventana.geometry(f"{screen_width}x{screen_height}")
# Crear un objeto Style
estilo = ttk.Style()
# Configurar el color de fondo de las pestañas
estilo.configure("Fondo.TFrame", background="lightblue")
```

Figura 21: Código para la implementación de la ventana.

Tal como se ha mencionado previamente, se dispone de dos modelos de predicción: uno para polaridad y otro para emociones. Con el objetivo de integrarlos en una única interfaz de forma clara y organizada, se optó por implementar dos pestañas o Frames, permitiendo que ambos modelos coexistan en la misma ventana y facilitando al usuario la navegación entre ellos. Para llevarlo a cabo se ha hecho uso de las siguientes clases:

- **Frame()**: contenedor que permite organizar y agrupar la disposición de los widgets dentro de la interfaz gráfica [42]. En la Figura 22 podemos ver el código utilizado para crear las pestañas Emotions y Polarity.

```
# Crear un widget Notebook (pestañas)
pestanas = ttk.Notebook(ventana)
# Crear el contenido de cada pestaña
frame1 = ttk.Frame(pestanas)
frame2 = ttk.Frame(pestanas)
# Agregar contenido a cada pestaña
pestanas.add(frame1, text="Emotions")
pestanas.add(frame2, text="Polarity")
# Empaquetar el Notebook
pestanas.pack(expand=1, fill="both")
```

Figura 22: Código para la implementación de las pestañas *Emotions* y *Polarity*.

- **Button()**: botón interactivo que ejecuta una función o comando cuando es presionado por el usuario [42].
- **Label()**: se utiliza para mostrar texto o imágenes no editables en la interfaz [42]. Como podemos ver en la Figura 23, lo hemos utilizado para añadir un logo a la interfaz.

```
try:
    imagen = Image.open('C:/Users/lucia/Desktop/TFG/Emociones/AI_MH_SN.jpeg')
    #imagen = Image.open(relative_path)
    imagen = imagen.resize((50, 50), resample=Image.BICUBIC)
    foto = ImageTk.PhotoImage(imagen)
    # Crear un label para mostrar la imagen
    label_imagen1 = tk.Label(frame1, image=foto, width=50, height=50)
    label_imagen1.place(x=1285, y=640)
    label_imagen2 = tk.Label(frame2, image=foto, width=50, height=50)
    label_imagen2.place(x=1285, y=640)
except FileNotFoundError:
    print("Error: No se encontró la imagen")
    exit()
```

Figura 23: Código para la implementación del logo MENTAI.

Tanto para los botones como para los cuadros de texto se han utilizado diferentes opciones para poder realizar una personalización completa de nuestra interfaz, como podemos ver en la Figura 24:

- **Text**: texto que aparece en el botón.
- **Command**: se utiliza para llamar al método cuando ese botón es presionado por el usuario.
- **Padding**: indica la separación horizontal y vertical entre el botón y el resto de los elementos.
- **State**: se utiliza para indicar el estado del objeto, en nuestro caso hemos utilizado Normal y Disabled.

```
#---------------------------------PARTE DE ARRIBA (Emociones --> frame1) ----------------------------

titulo_frame1 = ttk.Label(frame1, text="Insert a comment:")
titulo_frame1.place(x=50, y=40)
cuadro_texto_frame1 = ttk.Entry(frame1, width=30)
cuadro_texto_frame1.place(x=50, y=60)

titulo_prediccion_frame1 = ttk.Label(frame1, text="Comment prediction:")
titulo_prediccion_frame1.place(x=550, y=40)
cuadro_prediccion_frame1 = ttk.Entry(frame2, width=30, state="disabled")
cuadro_prediccion_frame1.place(x=550, y=60)

boton_algoritmo_frame1 = ttk.Button(frame1, text="Run Algorithm", command=boton_algoritmo, padding=(43, 10))
boton_algoritmo_frame1.place(x=1100, y=40)

#-----------------------------------------------PARTE DE ABAJO (Emociones) ----------------------------

botoncsv_frame1 = ttk.Button(frame1, text="Insert csv/Excel", command=cargar_archivo, padding=(40, 10), style='ColorBoton.TButton')
botoncsv_frame1.place(x=50, y=150)

boton_ejecut_csv_frame1 = ttk.Button(frame1, text="Run Algorithm with Csv/Excel", command=ejecutar_excel,
                                     padding=(10, 10), style='ColorBoton.TButton', state="disabled")
boton_ejecut_csv_frame1.place(x=550, y=150)

boton_descargar_csv_frame1 = ttk.Button(frame1, text="Download Excel/CSV", command=descargar_excel,
                                        padding=(43, 10), style='ColorBoton.TButton', state="disabled")

boton_descargar_csv_frame1.place(x=1100, y=150)
```

Figura 24: Código para la implementación de los botones en la pestaña *Emotions*.

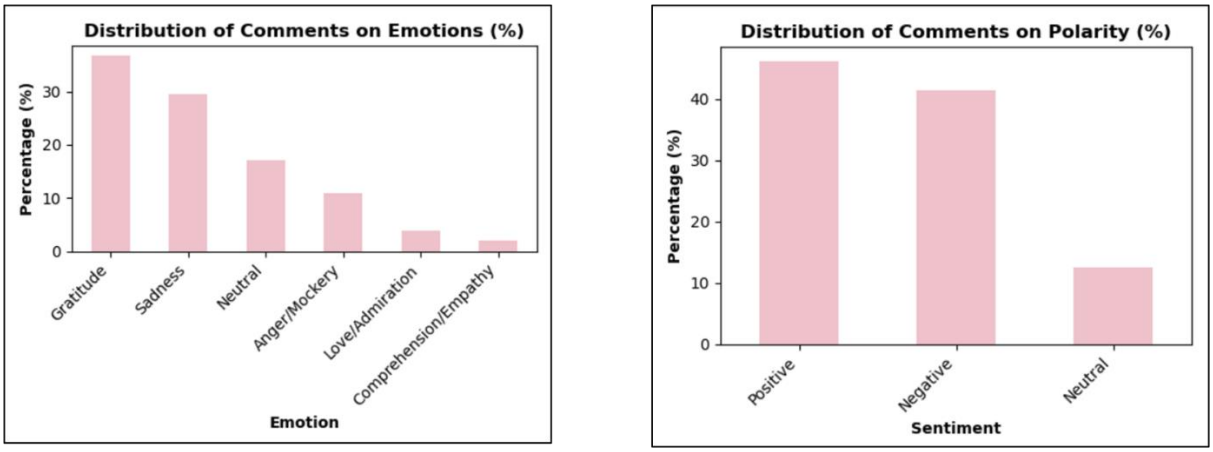
6.3.4 Creación de las gráficas

Para la generación de las gráficas se ha empleado el módulo matplotlib.pyplot, una colección de funciones que permite operar de forma similar a MATLAB [43]. En este proyecto, se ha utilizado para construir y personalizar visualmente los resultados obtenidos

tras la ejecución del algoritmo, facilitando una primera interpretación sin necesidad de exportar los datos a un fichero Excel.Entre las funciones utilizadas para configurar las gráficas se encuentran figure(), plot(), yticks(), title() y subplots_adjust(). Sin embargo, matplotlib.pyplot por sí solo no permite integrar las gráficas directamente en la interfaz. Para ello, se recurre a la clase FigureCanvasTkAgg, la cual habilita la visualización de las gráficas en una ubicación específica dentro de la interfaz construida con tkinter.

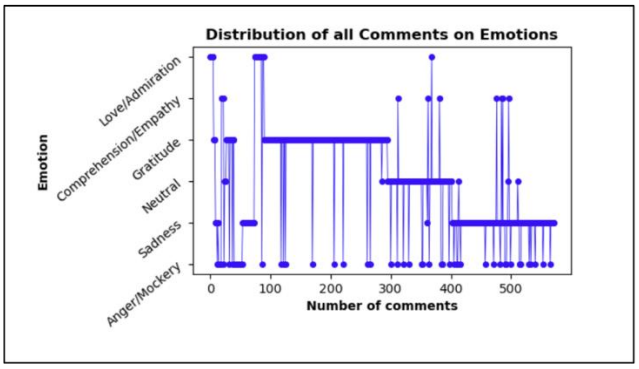
Todo esto se ha integrado dentro de las clases llamadas mostrar_gráfica1 y mostrar_gráfica2 para la pestaña de *Emotions* y en mostrar_gráfica3 y mostrar_gráfica4 para la pestaña de *Polarity*. A continuación, se va a explicar que se muestra en cada uno de los dos tipos gráficas que se han implementado:

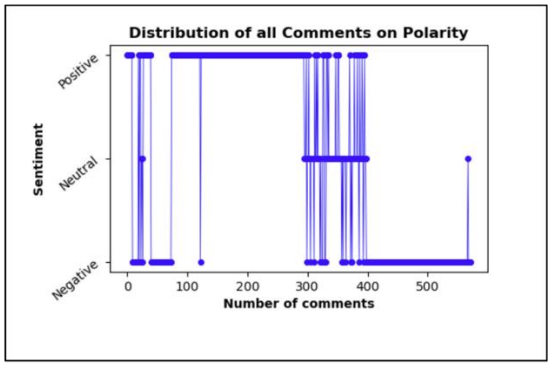
- **mostrar_grafica1** y **mostrar_gráfica3**: Estas gráficas de barras se encuentran en el lado izquierdo de las pestañas de *Emotions* y *Polarity* respectivamente. En ella se muestran las emociones/polaridades en el eje x y el número de comentaros de cada uno en el eje y expresado en tanto %. Así podemos ver rápidamente la cantidad de comentarios de cada tipo representado. En las Figuras 25 y 26 podemos ver un ejemplo de la gráfica para emociones y polaridades respectivamente.



Figuras 25, 26: Distribución de los comentarios de emociones y polaridades respectivamente.

- **mostrar_grafica2** y **mostrar_gráfica4**: Estas gráficas de evolución se encuentran en el lado derecho de las pestañas de *Emotions* y *Polarity* respectivamente. En ella se va mostrando la evolución de las emociones/polaridades del fichero precargado. En las Figuras 27 y 28 podemos ver un ejemplo de la gráfica para emociones y polaridades respectivamente.





Figuras 27,28: Evolución de los comentarios de emociones y polaridades respectivamente.

6.3.5 Diseño final y ejecución de la interfaz

En este apartado final se presentan de forma visual los resultados obtenidos tras la ejecución del código descrito en los apartados anteriores. Al ejecutar el código se genera una ventana emergente, tal como se ilustra en la Figura 29, para un sistema operativo Windows.

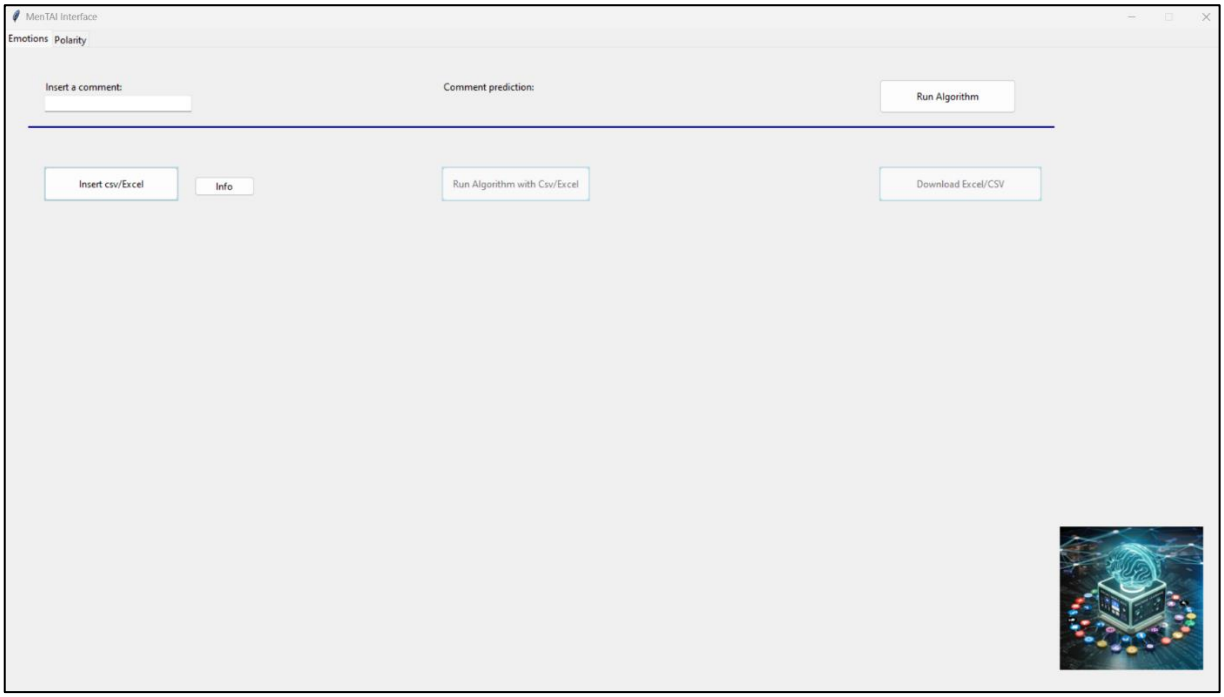


Figura 29: Interfaz gráfica en Windows.

En la parte superior de la interfaz se encuentran las pestañas *Emotions* y *Polarity*, que permiten al usuario seleccionar el tipo de predicción que desea realizar sobre los comentarios. Una vez elegido, vemos dos partes bien diferenciadas. En la parte superior tenemos la opción de introducir un comentario manualmente y obtener su predicción, como

se puede ver en la Figura 30. En ella tenemos un cuadro de texto con la indicación *Insert a comment*, en el cual el usuario puede introducir manualmente una palabra o frase. Para ejecutar la predicción, es necesario pulsar el botón *Run Algorithm*. A continuación, se mostrará la emoción o polaridad predicha —dependiendo de la pestaña activa— en el campo *Comment prediction*.

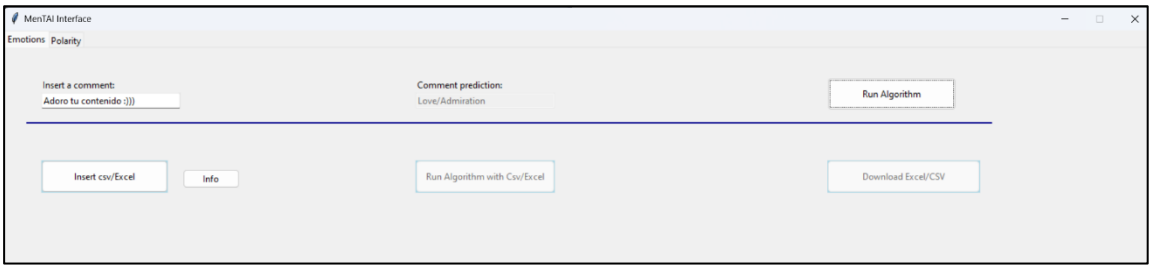


Figura 30: Resultado de la introducción de un comentario manualmente para emociones.

Debajo de la línea divisoria, nos permite cargar y procesar un fichero del cual posteriormente podremos obtener tanto las gráficas asociadas como la posibilidad de descargar las predicciones. Inicialmente, únicamente se encuentra habilitado el botón *Insert csv/Excel*, mientras que los otros dos permanecen desactivados. Al hacer clic sobre este botón, se despliega una ventana emergente que permite al usuario seleccionar el fichero que desea cargar.

Una vez que el fichero se ha cargado correctamente, se mostrará el mensaje *File uploaded succesfully* y se habilitará el botón *Run Algorithm with csv/Excel*. Al pulsarlo, se visualizarán en la interfaz las dos gráficas descritas en el apartado anterior. Esto permite obtener una primera representación visual de los resultados y verificar que el funcionamiento del sistema es el esperado. El resultado final será similar al que se muestra en la Figura 31. Finalmente se habilitará el botón *Download Excel/CSV* permitiéndonos así descargar los resultados obtenidos y poder realizar las acciones deseadas.

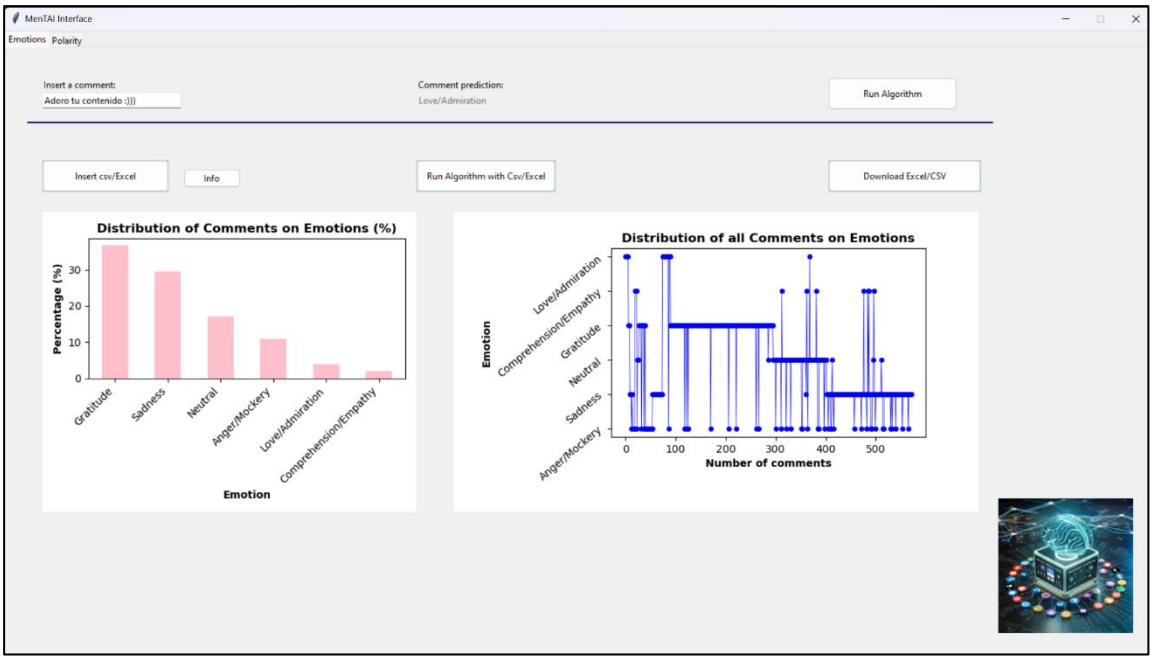


Figura 31: Resultado de la carga de un fichero para emociones.

6.4 Implementación como aplicación

Como parte de la integración final y con el objetivo de mejorar la accesibilidad de la herramienta, se llevó a cabo una colaboración con un compañero de titulación, autor de un Trabajo de Fin de Grado enfocado en el desarrollo de una aplicación de aprendizaje automático para la evaluación del lenguaje en plataformas como Twitch. Fruto de esta colaboración, se logró integrar la interfaz gráfica en un archivo ejecutable, lo que permite al usuario utilizar la herramienta de forma sencilla, sin necesidad de entornos de desarrollo ni conocimientos técnicos adicionales.

6.5 Conclusiones

En este capítulo se ha descrito detalladamente el proceso de desarrollo y funcionamiento de la interfaz gráfica. Esto permite valorar las múltiples posibilidades que ofrecen las bibliotecas de Python para la creación de entornos interactivos. Asimismo, se ha puesto especial énfasis en la importancia de diseñar una herramienta accesible, que permita a cualquier usuario utilizar el modelo sin necesidad de conocimientos en programación. La incorporación de gráficas para representar visualmente los resultados facilita un análisis rápido e intuitivo, aportando una visión inicial clara de las predicciones obtenidas. Además, con el objetivo de facilitar su distribución y uso, se ha implementado la interfaz en forma de aplicación ejecutable.

7

Conclusiones y líneas futuras

7.1 Conclusiones

El presente Trabajo de Fin de Grado se ha enfocado en la optimización de un modelo basado en la arquitectura BERT, RoBERTuito, entrenado para clasificar con la máxima precisión la polaridad y las emociones expresadas en los comentarios de los posts de Instagram seleccionados en el contexto de la salud mental.

Dicho entrenamiento se ha realizado con un nuevo corpus, se han añadido nuevos comentarios para tratar de equilibrar tanto las polaridades como las emociones. Los resultados obtenidos en la clasificación de polaridad alcanzan un 89,09 %, lo que representa una ligera disminución respecto al 93,3 % reportado en investigaciones anteriores. En cuanto a la clasificación de emociones, se logró un 85,58 %, muy cercano al 86 % obtenido en estudios previos. Este rendimiento similar, aunque levemente inferior, podría explicarse no solo por el aumento del volumen de comentarios analizados, sino también por la incorporación de publicaciones provenientes de contextos más variados y con mayor diversidad lingüística. La inclusión de expresiones vinculadas a distintos personajes públicos diversos introduce matices semánticos y usos del lenguaje que pueden generar ambigüedades, especialmente en términos que varían su carga emocional o polaridad según el contexto.

En cuanto a la interfaz, se ha verificado su correcto funcionamiento ejecutando el algoritmo, cargando datos en los diferentes formatos de ficheros disponibles e introduciendo un comentario manualmente. La herramienta genera correctamente gráficos que permiten visualizar directamente los resultados obtenidos y también tiene la opcion de poder descargar los comentarios con su predicción. Para garantizar su uso a nivel global, todos los componentes presentes en la interfaz se han implementado en inglés. Al estar

desarrollada en Python, se asegura su compatibilidad con distintos sistemas operativos, ofreciendo mínimas variaciones visuales.

7.2 Líneas futuras

Debido a que los resultados obtenidos no presentan la mejoría que cabría esperar, nos lleva a reflexionar sobre las mejoras a realizar en investigaciones posteriores. Se propone:

- **Ampliación y mejora del corpus**: Aunque para el presente trabajo ya se ha realizado una ampliación, sería conveniente incluir aún más comentarios e incidir específicamente en las clases que tienen menor representación. Además, sería muy útil elegir comentarios con la menor ambigüedad posible para evitar la confusión entre clases.
- **Optimización del modelo**: Mejorar la precisión especialmente para las emociones que han sido más difíciles de identificar y las clases de Neutral e Indeterminado.
- **Adaptación del modelo a contextos culturales diversos y expansión de categorías emocionales**: Se podrían desarrollar versiones adaptadas del modelo para comunidades que presenten estigmas culturales distintos, ampliando así su aplicabilidad y sensibilidad intercultural. Por ejemplo, sería relevante incorporar publicaciones de personajes hispanohablantes u otras figuras públicas pertenecientes a contextos socioculturales diversos, lo que permitiría captar variaciones lingüísticas, referencias culturales y formas particulares de expresión emocional. Asimismo, se podrían añadir categorías emocionales más complejas o matizadas que aporten información adicional, como emociones mixtas, niveles de intensidad o dimensiones contextuales (e.g., ironía, resignación, entusiasmo), mejorando así la riqueza interpretativa del modelo.
- **Mejora de la interfaz**: e plantea una optimización integral de la interfaz, tanto a nivel de rendimiento como de experiencia de usuario. Esto incluye la depuración y optimización del código para reducir significativamente los tiempos de procesamiento y respuesta, lo que permitiría una interacción más fluida y eficiente. Además, se propone el desarrollo de una interfaz web profesional, accesible desde distintos dispositivos, que ofrezca visualizaciones intuitivas, opciones de filtrado y exportación de resultados, así como un diseño adaptable y centrado en la usabilidad. Esta mejora facilitaría el acceso a usuarios no técnicos, ampliando el alcance de la herramienta en contextos académicos, institucionales o profesionales.

8

Bibliografía

[1] Yasin Dus, Georgiy Nefedov, “An Automated Tool to Detect Suicidal Susceptibility from Social Media Posts” Cornell University, 2024. https://doi.org/10.48550/arXiv.2310.06056

[2] J. E. Asensio, N. M. Alvarez, y J. M. V. Hernández, “Análisis emocional en redes sociales basados en modelos de aprendizaje automático transformers BERT”, Universidad de Valladolid, 2023. https://uvadoc.uva.es/handle/10324/62911.

[3] M. C. Lanchares y N. M. Álvarez, “Aplicación de aprendizaje automático para evaluar lenguaje de videojuegos en Twitch”, Universidad de Valladolid, 2024. https://uvadoc.uva.es/handle/10324/71264

[4] S. B, P. R. P, S. M. B and K. S, "The Evolution of Large Language Model: Models, Applications and Challenges," 2024 International Conference on Current Trends in Advanced Computing (ICCTAC), Bengaluru, India, 2024, pp. 1-8, doi: 10.1109/ICCTAC61556.2024.10581180
https://ieeexplore.ieee.org/document/10581180/authors#authors

[5] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

[6] Rasley, J., Rajbhandari, S., Ruwase, O., & He, Y. (2020, August). Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp.3505-3506).

[7] Rajbhandari, S., Rasley, J., Ruwase, O., & He, Y. (2020, November). Zero: Memory optimizations toward training trillion parameter models. In SC20: International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-16). IEEE

[8] He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., & Neubig, G. (2021). Towards a unified view of parameter-efficient transfer learning. arXiv preprint arXiv:2110.04366.

[9] J. M. Pérez, D. A. Furman, L. Alonso Alemany, y F. M. Luque, "RoBERTuito: a pre-trained language model for social media text in Spanish", en Proceedings of the Thirteenth Language Resources and Evaluation Conference, 2022, pp. 7235–7243. Disponible en: https://aclanthology.org/2022.lrec-1.785/.

[10] Organización Mundial de la Salud (2023) Acción acelerada mundial en favor de la salud de los adolescentes (AA-HA!): orientación para apoyar la implementación en los países. Organización Mundial de la Salud, Ginebra

[11] Fonseca-Pedrero E, Calvo P, Díez-Gómez A, Pérez-Albéniz A, Lucas-Molina B, Al-Halabí S (2023) La salud mental de los adolescentes en contextos educativos: reflexiones derivadas del estudio PSICE. Consejo General de la Psicología de España, Madrid

[12] Unicef (2020) Salud Mental en la infancia en el Escenario de la COVID-19. Propuestas de Unicef España. UNICEF España, Madrid

[13] Hendrikse C, Limniou M (2024) El uso de Instagram y TikTok en relación con el uso problemático y el bienestar. J Technol Behav Sci. https://doi.org/10.1007/s41347-024-00399-6

[14] IAB España (2023) Estudio de Redes Sociales 2023 https://iabspain.es/estudio/estudio-de-redes-sociales-2023/

[15] https://www.thelancet.com/journals/lanpub/article/PIIS2468-2667(24)00125-7/fulltext

[16]https://www.nature.com/articles/s41598-024-83951-x?fromPaywallRec=false

[17] https://link.springer.com/article/10.1007/s00787-022-02012-8

[18] https://www.nature.com/articles/d41586-023-00402-9

[19] https://jamanetwork.com/journals/jamapediatrics/fullarticle/2799812

[20] https://www.who.int/es/news-room/fact-sheets/detail/adolescent-mental-health

[21] https://www.hhs.gov/es/surgeongeneral/reports-and-publications/youth-mental-health/social-media/index.html

[22] https://www.ontsi.es/es/publicaciones/Impacto-del-uso-de-Internet-y-redes-sociales-salud-mental-jovenes-adolescentes

[23] Python. Disponible en: https://www.python.org.

[24] Pandas. Disponible en: https://pandas.pydata.org/.

[25] Numpy. Disponible en: https://numpy.org/.

[26] Matplotlib.pyplot. Disponible en: https://matplotlib.org/.

[27] Hugging Face Transformers. Disponible en: https://huggingface.co/docs/transformers/index.

[28] Google colab. Disponible en: https://colab.research.google.com/

[29] Anaconda. Disponible en: https://www.anaconda.com/

[30] Jupyter Notebook. Disponible en: https://jupyter.org/

[31] Wandb. Disponible en: https://wandb.ai/site

[32] Instagram. Disponible en: https://www.instagram.com/

[33] ¿Cómo aprovechar el rendimiento de la matriz de confusión? Disponible en: https://datascientest.com/es/matriz-de-confusion

[34] J. M. Pérez, D. A. Furman, L. Alonso Alemany, y F. M. Luque, "RoBERTuito: a pre-trained language model for social media text in Spanish", en Proceedings of the

Thirteenth Language Resources and Evaluation Conference, 2022, pp. 7235–7243. Disponible en: https://aclanthology.org/2022.lrec-1.785/.

[35] J. M. Pérez et al., “pysentimiento: A Python Toolkit for Opinion Mining and Social NLP tasks”, arxiv.org, 2021. Disponible en: https://arxiv.org/abs/2106.09462.

[36] M. Sussmann, “What are the differences between artificial intelligence, machine learning, deep learning and generative AI?”, sumologic.com, 23-abr-2024. Disponible en: https://www.sumologic.com/blog/machine-learning-deep-learning/.

[37] “Tokenizer”, Huggingface.co. Disponible en: https://huggingface.co/docs/transformers/main_classes/tokenizer.

[38] Hugging Face Transformers. Disponible en: https://huggingface.co/docs/transformers/index.

[39] Token type ids, “Glossary”, Huggingface.co. Disponible en: https://huggingface.co/docs/transformers/glossary#tokentype-ids.

[40] Weights & Biases. Disponible en:https://wandb.ai/site.

[41] Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.

[42] Python Software Foundation. (s.f.). *tkinter — Interfaz de Python para Tcl/Tk*. Documentación de Python 3. https://docs.python.org/es/3/library/tkinter.html

[43] J. D. Hunter, M. Droettboom y T. Caswell, "Matplotlib 2.0.2 Documentation," Matplotlib, 2017. [En línea]. Disponible en: https://matplotlib.org/2.0.2/index.html