

# Sistema de Recomendaciones de Hoteles Basado en Análisis de Emociones

Miguel Alejandro Yáñez Martínez y Darío Rodríguez Llosa

MATCOM, Universidad de La Habana

**Resumen:** Este proyecto desarrolla un sistema de recomendaciones de hoteles, que mejora la experiencia de los usuarios al buscar alojamiento, utilizando para ello el análisis de emociones expresadas en las reseñas de los clientes y evaluando la satisfacción del cliente a partir de dichas reseñas. Se aplicó un modelo de reconocimiento de emociones entrenado específicamente para esta tarea, lo que permite identificar de manera más precisa las emociones positivas, neutras y negativas. Los resultados obtenidos ayudan a clasificar los hoteles en función de la satisfacción de los huéspedes, lo que facilita al usuario elegir el hotel más adecuado según su percepción de calidad.

Palabras Claves: BERT · Procesamiento del Lenguaje Natural · Sistema de Recomendaciones · Hoteles.

## Repositorio:

<https://github.com/Matcom-Projects/Hotel-Recommendation-System>

## 1 Introducción

El análisis de comentarios en la industria hotelera ha cobrado relevancia debido a su impacto directo en la percepción y toma de decisiones de los clientes. En este contexto, el reconocimiento de emociones en el análisis de texto ha demostrado ser una herramienta eficaz para capturar la complejidad de las respuestas emocionales de los usuarios. A diferencia de los enfoques tradicionales de análisis de sentimientos, que se limitan a clasificar los comentarios en categorías básicas como "positivos" o "negativos", el reconocimiento de emociones permite identificar y cuantificar de manera más precisa las diversas emociones expresadas en las reseñas.

Este trabajo desarrolla un sistema de reconocimiento de emociones diseñado para analizar comentarios sobre hoteles y evaluar la positividad de los mismos. A partir de esta evaluación, se genera un ranking de hoteles basado en la intensidad y frecuencia de emociones positivas identificadas en las reseñas de los usuarios. Esta metodología no solo ofrece una visión más detallada del grado de satisfacción de los clientes, sino que también proporciona un mecanismo objetivo para clasificar los hoteles según la calidad percibida por sus huéspedes.

## 2 Estado del Arte

El análisis de sentimientos es una rama de la inteligencia artificial <sup>[5]</sup> y el procesamiento del lenguaje natural <sup>[6]</sup> que tiene sus raíces en la necesidad humana de comprender las emociones y opiniones expresadas en el lenguaje. Desde tiempos antiguos, los filósofos y teóricos del lenguaje han explorado cómo las palabras pueden reflejar estados emocionales. Sin embargo, fue con el advenimiento de la computación en el siglo XX que se comenzaron a desarrollar técnicas más sistemáticas para analizar el lenguaje.

En las décadas de 1950 y 1960, los primeros intentos de análisis de sentimientos <sup>[7]</sup> se centraron en métodos manuales y heurísticos. Los investigadores utilizaban listas de palabras positivas y negativas para clasificar textos. Este enfoque rudimentario era laborioso y limitado, ya que no podía captar matices o contextos complejos. A medida que la tecnología avanzaba, surgieron nuevas metodologías que incorporaban reglas gramaticales y sintácticas para mejorar la precisión del análisis.

Con el auge de Internet en los años 90 y 2000, la cantidad de datos textuales disponibles aumentó exponencialmente. Esto llevó al desarrollo de técnicas más sofisticadas basadas en aprendizaje automático. Algoritmos como Naïve Bayes <sup>[8]</sup> y máquinas de soporte vectorial (SVM) <sup>[9]</sup> comenzaron a ser utilizados para clasificar textos según su polaridad emocional. Estos métodos permitieron un análisis más eficiente y preciso, aunque aún requerían una considerable cantidad de datos etiquetados para entrenar los modelos.

La llegada del aprendizaje profundo <sup>[10]</sup> en la última década ha revolucionado el campo del análisis de sentimientos. Modelos como las redes neuronales recurrentes (RNN) <sup>[11]</sup> y transformadores <sup>[1]</sup> <sup>[3]</sup> han demostrado ser altamente efectivos para capturar patrones complejos en el lenguaje. Herramientas como BERT (Bidirectional Encoder Representations from Transformers) <sup>[13]</sup> han permitido a los sistemas entender mejor el contexto y las sutilezas del lenguaje humano, lo que ha llevado a mejoras significativas en la precisión del análisis.

Hoy en día, el análisis de sentimientos se aplica en diversas áreas, desde marketing hasta atención al cliente, ayudando a las empresas a comprender mejor las opiniones de sus clientes y a tomar decisiones informadas. A medida que la tecnología continúa evolucionando, es probable que veamos aún más avances en esta fascinante disciplina, permitiendo una comprensión más profunda de las emociones humanas expresadas a través del lenguaje escrito.

### 2.1 Diferentes enfoques de Análisis de Sentimientos

El análisis de sentimientos es un campo crucial dentro del procesamiento del lenguaje natural (NLP), que se centra en identificar y extraer opiniones subjetivas de textos. Este proceso se puede clasificar en dos enfoques principales: **basados en reglas** y **automáticos** <sup>[15]</sup>. Cada uno de estos enfoques utiliza diferentes técnicas y metodologías, lo que influye en su efectividad y aplicabilidad.

**Enfoques Basados en Reglas:** Los sistemas basados en reglas utilizan un conjunto de directrices manuales para determinar la subjetividad, polaridad o tema de una oración. Herramientas como expresiones regulares ayudan a identificar patrones en el texto. Dentro de este enfoque, se aplican técnicas como:

- **Preprocesamiento:** Técnicas como la tokenización, stemmatización y análisis sintáctico <sup>[12]</sup> son clave para dividir el texto en unidades significativas.
- **Léxicos:** Utilizan listas de palabras polarizadas (positivas y negativas) para evaluar el sentimiento en un texto.

Estos enfoques son efectivos para tareas simples, pero fallan en captar ironías, sarcasmos o contextos complejos.

**Enfoques Automáticos:** Los métodos automáticos, basados en aprendizaje automático, no dependen de reglas predefinidas, sino que aprenden de ejemplos etiquetados. Esto permite mayor flexibilidad ante variaciones lingüísticas. Los principales algoritmos utilizados son:

- **Naïve Bayes:**<sup>[8]</sup> Es un clasificador probabilístico que asume independencia entre características.
- **Regresión Logística:** <sup>[16]</sup> Es un modelo lineal que predice el sentimiento en función de variables independientes.
- **Máquinas de Vectores de Soporte (SVM):**<sup>[9]</sup> Es un algoritmo supervisado que busca el mejor hiperplano para separar clases.
- **Aprendizaje Profundo:**<sup>[10]</sup> Son redes neuronales que aprenden representaciones complejas a partir de grandes volúmenes de datos.

El aprendizaje profundo, especialmente con modelos basados en transformers, ha revolucionado el análisis de sentimientos al capturar relaciones contextuales más complejas. Esto permite una mayor precisión y adaptabilidad frente a la complejidad del lenguaje humano.

**Transformers:** Los transformers son una arquitectura de redes neuronales introducida por primera vez en el artículo "Attention Is All You Need"<sup>[1]</sup> en 2017 por Vaswani et al., suponiendo una revolución en el ámbito del procesamiento del lenguaje natural. Una de las características más destacadas de esta arquitectura es su mecanismo de atención, que le permite utilizar la información de contenidos relevantes en el texto de entrada al generar la salida, sin importar la distancia entre dichos contenidos. En algoritmos generativos de texto, al generar una palabra a la vez, el modelo es capaz de utilizar sus propios resultados como datos de entrada, lo que dota al resultado final de mayor coherencia.

En la figura se puede observar cómo, a partir del texto "As aliens entered our planet", se genera una continuación de la historia palabra por palabra, así como las palabras a las que se presta atención para la generación de la siguiente.

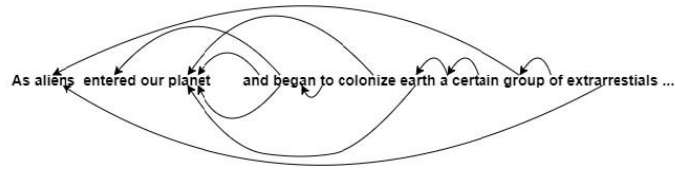


Fig. 1: Mecanismo de atención de un texto generativo

Las redes neuronales recurrentes (RNN) también poseen capacidades de "atención" a entradas anteriores; sin embargo, su mecanismo no es tan eficiente debido a la pérdida de memoria. En el supuesto anterior, a medida que la historia se alarga, llegará un momento en que la ventana de entradas a las que una RNN podría hacer referencia sería parcial, mientras que para los transformers seguiría siendo completa.

En la figura se muestra la estructura habitual de un modelo transformer:

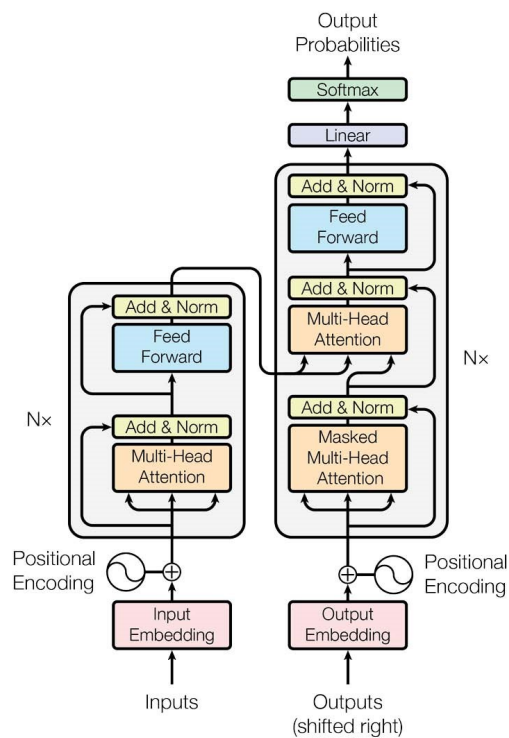


Fig. 2: Estructura de un modelo transformer

Dicha estructura se desglosa en los siguientes elementos:

- **Input embedding:** Se encarga de crear un vector de representación para cada palabra de los datos de entrada, presentando los datos a la red en forma numérica. Estos vectores pueden considerarse como una tabla de referencias y este elemento también se conoce como tokenizador en el ámbito de este trabajo.
- **Capa codificadora:** Compuesta por la agrupación del margen izquierdo, cuyo objetivo es crear una representación continua a partir de la secuencia de entradas que contenga toda la información aprendida sobre atención relativa a la secuencia completa. Esta capa puede repetirse varias veces para obtener nuevas representaciones y potenciar los resultados.
- **Capa decodificadora:** Formada por la agrupación del margen derecho, parte desde un token inicial junto con ciertas palabras de la secuencia de entrada y las salidas de la capa codificadora para generar un conjunto final. Estas salidas pasan por una capa lineal que actúa como clasificador con un número igual al total de palabras disponibles en el modelo. Posteriormente, estas salidas son procesadas por una capa softmax final, obteniendo probabilidades para cada uno de los resultados; así, la salida con mayor probabilidad será considerada como la predicción del modelo.

Estos modelos han superado ampliamente enfoques anteriores en diversas tareas, entre las cuales destacan generación de texto, predicción de palabras ocultas, traducción, etiquetado de entidades y clasificación de sentimientos, tarea en la que se centra este trabajo. A partir del modelo transformer han surgido extensiones como Generative Pre-trained Transformer (GPT) <sup>[14]</sup>, utilizado en el popular asistente ChatGPT, T5 (Text-to-Text Transfer Transformer) y BERT, del cual hablaremos a continuación.

**BERT:** (Bidirectional Encoder Representations from Transformers) es una extensión del modelo transformer creada por investigadores de Google AI en 2018.

BERT introdujo nuevas tareas de preentrenamiento: el Modelo de Lenguaje Enmascarado (MLM) y la Predicción de la Siguiente Oración (NSP). El MLM enmascara el 15% de las palabras aleatorias en una oración, reemplazándolas con un token especial como [MASK], y entrena al modelo para predecir esas palabras en función de su contexto bidireccional. Esta tarea permite a BERT capturar mejor el significado contextual tanto de las palabras anteriores como posteriores, lo que resulta en representaciones más ricas y precisas. Esta bidireccionalidad contrasta con modelos unidireccionales como GPT, que solo pueden analizar el contexto previo de una palabra, limitando su comprensión del texto.<sup>[18]</sup>

Predicción de la Siguiente Oración (NSP) es otra tarea clave en BERT. El modelo recibe dos oraciones y debe predecir si la segunda oración sigue lógicamente a la primera. El 50% de las veces, las dos oraciones están en secuencia correcta, y el 50% restante, la segunda oración es seleccionada aleatoriamente de otra parte del corpus. Esta tarea permite a BERT aprender relaciones contextuales de

largo alcance entre oraciones y mejorar su comprensión de la coherencia textual, algo que es crucial para tareas como la generación de resúmenes, respuestas a preguntas y comprensión de lectura.<sup>[18]</sup>

Respecto al resto de características no difiere significativamente respecto a otros modelos como GPT. Esta decisión fue intencionada ya que consideraron OpenAI GPT como el modelo más comparable a BERT, lo cual permitió ofrecer comparativas entre ambos modelos. Mientras que GPT fue entrenado con el dataset BooksCorpus, compuesto por 800 millones de palabras y utiliza una tasa uniforme para cualquier tarea específica. BERT fue entrenado combinando este dataset con Wikipedia completa y emplea diferentes tasas para cada tarea. En este trabajo utilizaremos modelos creados a partir o derivados del enfoque BERT publicados en la comunidad Hugging Face.

Un gran punto a favor de BERT es su capacidad para adaptarse a diferentes tareas dentro del procesamiento natural del lenguaje. Este modelo se entrena inicialmente con grandes cantidades de datos no etiquetados para aprender representaciones efectivas tanto para palabras como para frases. Posteriormente, puede ajustarse mediante conjuntos etiquetados específicos para realizar un fine-tuning hacia tareas concretas.

### **3 Entrenamiento del Modelo BERT para Análisis de Emociones<sup>[2]</sup>**

Para llevar a cabo el análisis de emociones, se eligió utilizar un modelo BERT, debido a su capacidad para comprender el contexto del lenguaje de manera profunda, su flexibilidad para adaptarse a tareas específicas y su rendimiento superior en comparación con otros modelos disponibles. Se utilizó el modelo preentrenado 'bert-base-uncased' <sup>[23]</sup>, que es una variante de BERT que diferencia entre letras mayúsculas y minúsculas. Este modelo ha sido previamente entrenado en un amplio corpus de texto, lo que le permite comprender las relaciones contextuales entre palabras y captar matices del lenguaje. Sin embargo, dado que el objetivo es identificar emociones específicas en reseñas de hoteles, es necesario realizar un proceso conocido como "fine-tuning" para aumentar la exactitud de los resultados.

El fine-tuning consiste en ajustar el modelo preentrenado a un conjunto de datos específico relacionado con la tarea de análisis emocional. Para ello, se utiliza un conjunto de datos etiquetados de reseñas de hoteles, que contiene ejemplos de texto junto con sus respectivas emociones <sup>[17]</sup>. Durante esta fase de ajuste, el modelo se entrena en este nuevo conjunto de datos durante varias épocas. Este proceso permite al modelo aprender patrones y características relevantes para la clasificación emocional.

Es fundamental realizar este fine-tuning porque, aunque BERT posee un conocimiento general del lenguaje, no está especializado en la identificación de emociones, especialmente en reseñas hoteleras. Al adaptar el modelo a las particularidades del lenguaje emocional presente en el conjunto de datos, se logra mejorar significativamente su rendimiento en esta tarea específica.

El objetivo principal es entrenar un modelo BERT para la tarea de clasificación de secuencias, donde las entradas de texto se clasifican en tres categorías, 'Negativa', 'Neutras' o 'Positivas', ya que estas son las clasificaciones utilizadas por el modelo. El proceso involucra la preparación de los datos, la configuración del modelo preentrenado, su entrenamiento con un conjunto de datos, la evaluación del modelo y la comparación con otros modelos, tanto los originales como versiones mejoradas.

### 3.1 Preprocesamiento de datos

Debido a que para el entrenamiento los datos de entrada debían presentar un formato específico, se procedió a modificar los datos. En primer lugar, se limpió el texto eliminando caracteres no deseados, como emojis, signos de puntuación y números. Se eliminaron las filas con valores nulos y se seleccionaron las columnas más relevantes para el análisis, como la dirección del establecimiento, las categorías, la ciudad, el país, el nombre del establecimiento, la provincia, la calificación de la reseña, el texto de la reseña y la etiqueta que indica el sentimiento (positivo, neutral o negativo). Esta limpieza garantiza que el análisis posterior se realice sobre datos consistentes y sin valores ausentes que puedan afectar los resultados.

Se codificaron las etiquetas de las reseñas según sus calificaciones, categorizándolas como 'Negativa', 'Neutras' o 'Positivas', ya que estas son las clasificaciones utilizadas por el modelo. Posteriormente, se tokenizaron las reseñas y se calculó la longitud de los tokens para cada una de las reseñas, con el fin de verificar si la cantidad de tokens excedía el tamaño máximo permitido por BERT (512 tokens). Los datos que se le pasan al modelo son las reseñas limpias y preprocesadas y las etiquetas numéricas correspondientes a las calificaciones sentimentales.

Se crearon dos conjuntos de datos, uno para el entrenamiento (90% de los datos) y otro para la evaluación (10% de los datos). Estos dos conjuntos de datos son fundamentales para entrenar un modelo BERT para tareas como el análisis de emociones o clasificación sentimental.

Finalmente, se realizó un estudio de estadísticas relevantes, como la distribución de las etiquetas, la longitud de los tokens y la cantidad de caracteres por reseña, entre otros, para comprender mejor la naturaleza de los datos. Los resultados del mismo aparecen en la Sección 'Resultados del Preprocesamiento de datos'

### 3.2 Entrenamiento

Debido al alto costo asociado con el entrenamiento de un modelo, se optó por realizar este proceso en Google Colab <sup>[4]</sup>, ya que proporciona acceso a recursos de computación en la nube, especialmente GPUs, que facilitan el entrenamiento del modelo. Durante la fase de entrenamiento, se llevó a cabo el ajuste de los parámetros del modelo a lo largo de varias épocas utilizando un optimizador y un programador de tasa de aprendizaje. Al final de cada época, se realizó la retropropagación y se actualizaron los parámetros del modelo, los cuales fueron

guardados posteriormente. Además, se evaluó el rendimiento del modelo en un conjunto de validación.

Para evitar el sobreajuste (overfitting), se seleccionó la versión del modelo que ofreciera los mejores resultados sin mostrar una disminución en el f1-score ya que indicaría la presencia de sobreajuste.

Finalmente, se realizó un análisis de errores en el conjunto de validación, calculando diversas métricas de rendimiento y generando una matriz de confusión. El rendimiento del modelo entrenado se comparó con el modelo BERT clásico utilizado para clasificación, así como con otro modelo previamente entrenado para el reconocimiento de emociones en reseñas, evaluando las métricas y las matrices de confusión de ambos modelos.

**Configuración del Modelo:** El modelo utilizado es una versión preentrenada de BERT con una capa adicional para la clasificación de secuencias. Esta capa final tiene tres salidas, correspondientes a las categorías a las que se desea clasificar el texto. Se inicializó un optimizador llamado AdamW, que es una versión modificada del algoritmo Adam para manejar mejor la regularización del modelo. Además, se configuró una tasa de aprendizaje que disminuye linealmente con el paso de las épocas para ayudar a estabilizar el entrenamiento.

Se establece también una configuración que define el número de épocas, la tasa de aprendizaje inicial, y el dispositivo en el que se ejecutará el entrenamiento, ya sea GPU o CPU.

**Entrenamiento del Modelo:** El entrenamiento se realizó durante cinco épocas. En cada época, el modelo pasa por todos los lotes de datos de entrenamiento. Durante este proceso, el modelo ajusta sus pesos en función de la diferencia entre las predicciones y las etiquetas verdaderas, medida a través de la función de pérdida.

En cada lote, el modelo recibe las secuencias de entrada junto con sus máscaras de atención y las etiquetas. Se realiza una pasada hacia adelante para obtener las predicciones y calcular la pérdida. Luego, se hace una pasada hacia atrás para calcular los gradientes, los cuales son utilizados para actualizar los parámetros del modelo. Finalmente, la tasa de aprendizaje es ajustada utilizando un scheduler para evitar que los ajustes sean demasiado grandes en las últimas etapas del entrenamiento.

La evaluación del modelo se lleva a cabo utilizando el conjunto de validación después de cada época. Durante la evaluación, el modelo se pone en modo de evaluación, donde no se calculan los gradientes y el objetivo es medir qué tan bien predice las etiquetas correctas.

En cada evaluación, se calculan las predicciones del modelo y se comparan con las etiquetas reales. Las predicciones y etiquetas son evaluadas utilizando la puntuación F1 ponderada, que considera tanto la precisión como el recall. Los resultados de evaluación muestran una disminución en la calidad del modelo a medida que pasa de la tercera a la quinta época, lo que confirma el problema de sobreajuste mencionado previamente.



Los resultados del entrenamiento muestran una mejora progresiva en la pérdida de entrenamiento a medida que avanzan las épocas, lo que indica que el modelo está aprendiendo a predecir mejor las etiquetas correctas. Los valores de la pérdida y la métrica F1 ponderada (una medida de precisión combinada con recall) para cada época son los siguientes:

- **Época 1**:
  - Pérdida de entrenamiento: 0.5549
  - Pérdida de validación: 0.5563
  - Puntuación F1 ponderada: 0.7812

- **Época 2**:
  - Pérdida de entrenamiento: 0.4613
  - Pérdida de validación: 0.6022
  - Puntuación F1 ponderada: 0.7888

- **Época 3**:
  - Pérdida de entrenamiento: 0.3797
  - Pérdida de validación: 0.7643
  - Puntuación F1 ponderada: 0.8007

- **Época 4**:
  - Pérdida de entrenamiento: 0.2887
  - Pérdida de validación: 0.9679
  - Puntuación F1 ponderada: 0.7941

- **Época 5**:
  - Pérdida de entrenamiento: 0.2241
  - Pérdida de validación: 1.1784
  - Puntuación F1 ponderada: 0.7802

En general, se observa que la pérdida de entrenamiento disminuye con el tiempo, lo que indica que el modelo está ajustándose a los datos de entrenamiento. Sin embargo, la pérdida de validación comienza a aumentar después de la tercera época, lo que sugiere que el modelo podría estar sobreajustándose (overfitting), es decir, aprendiendo detalles específicos de los datos de entrenamiento que no generalizan bien a datos no vistos.

Debido a que la pérdida de validación comienza a aumentar después de la tercera época, se eligió el estado alcanzado en la tercera época (**F1 ponderada de 0.8007**), ya que balancea un buen rendimiento en los datos de entrenamiento y validación.

**Resultados del entrenamiento:** Una vez entrenado el modelo, se realizó una comparación entre el modelo ajustado, el modelo original preentrenado de BERT sin ajuste fino y un modelo profesional.

Se realizaron pruebas de rendimiento de tres versiones diferentes de un modelo BERT entrenado para la clasificación de secuencias en tres categorías: "Negativo", "Neutro", y "Positivo".

Los modelos evaluados son los siguientes:

- **Modelo Ajustado (Fine-Tuned):** Modelo BERT ajustado en tareas específicas.
- **Modelo Original (Classic):** Modelo preentrenado sin ajuste fino.
- **Modelo Profesional:** Un modelo BERT preentrenado con más capas de ajuste específico para tareas de análisis de sentimientos.

El modelo profesional 'nptown/bert-base-multilingual-uncased' [22] es una variante del popular modelo BERT que ha sido ajustada específicamente para el análisis de sentimientos en reseñas de productos en seis idiomas: inglés, neerlandés, alemán, francés, español e italiano. Este modelo se caracteriza por su capacidad para comprender y procesar texto en múltiples lenguas, lo que lo convierte en una herramienta versátil para aplicaciones globales. Este modelo es un modelo general, que también puede ser sometido a un ajuste adicional (fine-tuning) para tareas relacionadas con el análisis de sentimientos específicos, como el que estamos haciendo nosotros. Se evalúan las tres versiones del modelo en el conjunto de validación y se analizan métricas como la precisión, el recall, la puntuación F1, y la exactitud general.

Table 1: Métricas de Clasificación Modelo Fine-Tune

	Precisión	Recall	F1-Score	Soporte
0	0.76	0.71	0.74	718
1	0.47	0.43	0.45	689
2	0.89	0.91	0.90	2919
<b>Exactitud</b>				4326
<b>Macro Avg</b>	0.70	0.69	0.70	4326
<b>Weighted Avg</b>	0.80	0.80	0.80	4326

Table 2: Métricas de Clasificación Modelo original

	Precisión	Recall	F1-Score	Soporte
0	0.34	0.04	0.08	718
1	0.17	0.00	0.01	689
2	0.68	0.98	0.81	2919
<b>Exactitud</b>				4326
<b>Macro Avg</b>	0.40	0.34	0.30	4326
<b>Weighted Avg</b>	0.54	0.67	0.56	4326

Table 3: Métricas de Clasificación Modelo profesional

<b>Clase</b>	<b>Precisión</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Soporte</b>
Negativo	0.17	0.92	0.29	718
Neutro	0.12	0.05	0.07	689
Positivo	0.70	0.03	0.07	2919
<b>Promedio Macro</b>	0.33	0.33	0.14	4326
<b>Promedio Ponderado</b>	0.52	0.18	0.10	4326

Table 4: Matriz de Confusión Modelo Fine-Tune

	<b>Predicho Negativo</b>	<b>Predicho Neutral</b>	<b>Predicho Positivo</b>
<b>Real Negativo</b>	513	138	67
<b>Real Neutral</b>	111	299	279
<b>Real Positivo</b>	52	200	2667

Table 5: Matriz de Confusión Modelo original

	<b>Predicho Negativo</b>	<b>Predicho Neutral</b>	<b>Predicho Positivo</b>
<b>Real Negativo</b>	31	2	685
<b>Real Neutral</b>	28	3	658
<b>Real Positivo</b>	33	13	2873

Table 6: Matriz de Confusión Modelo Profesional

	<b>Negativo</b>	<b>Neutro</b>	<b>Positivo</b>
<b>Negativo</b>	660	40	18
<b>Neutro</b>	628	35	26
<b>Positivo</b>	2602	215	102

A partir de los resultados obtenidos, se realizó una comparación cuantitativa entre el modelo ajustado y el original. Las diferencias observadas son las siguientes:

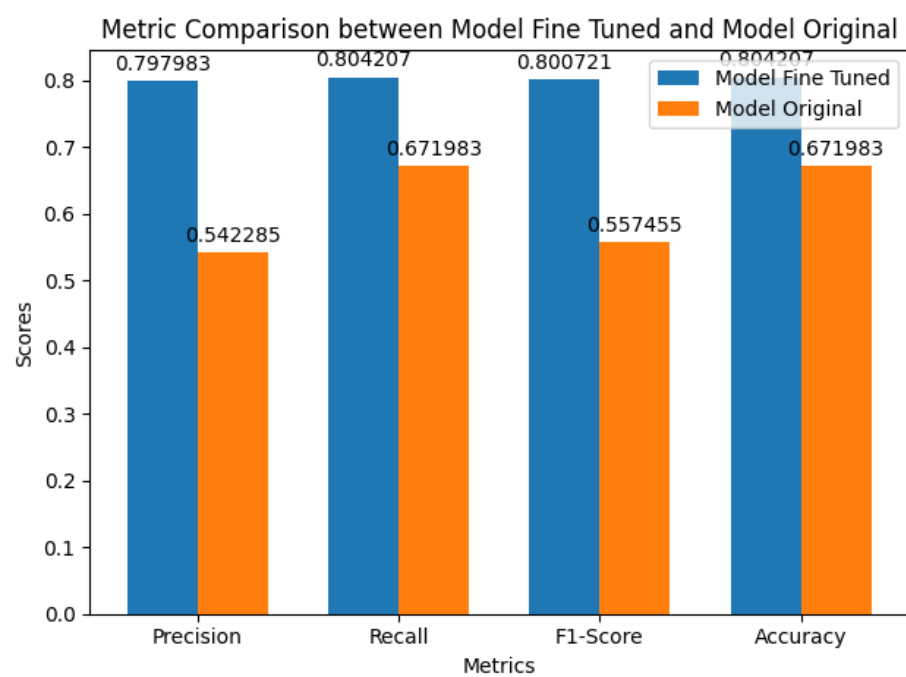


Fig. 3: Comparación métrica modelo fine-tune y original

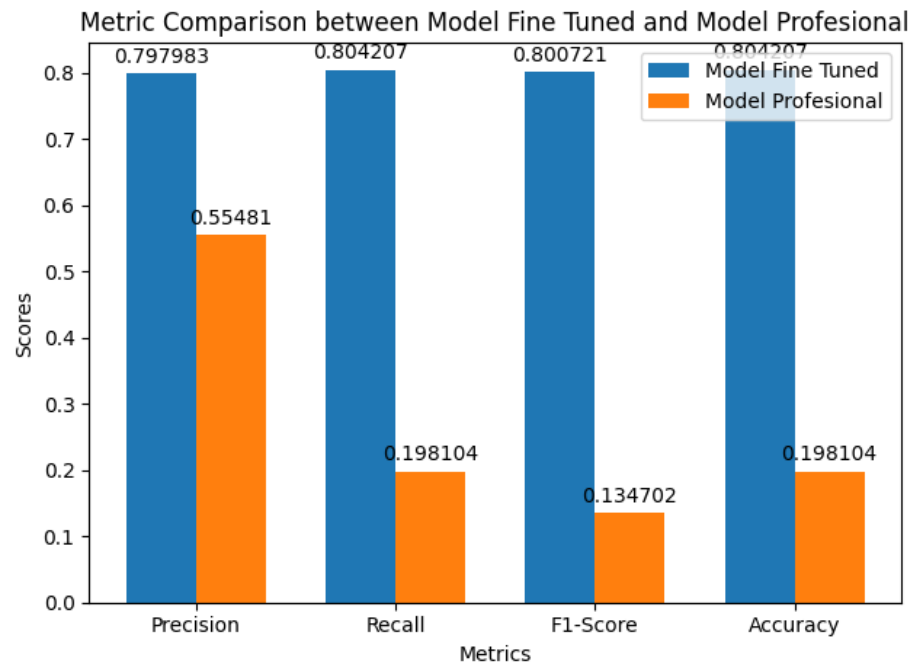


Fig. 4: Comparación métrica modelo fine-tune y profesional

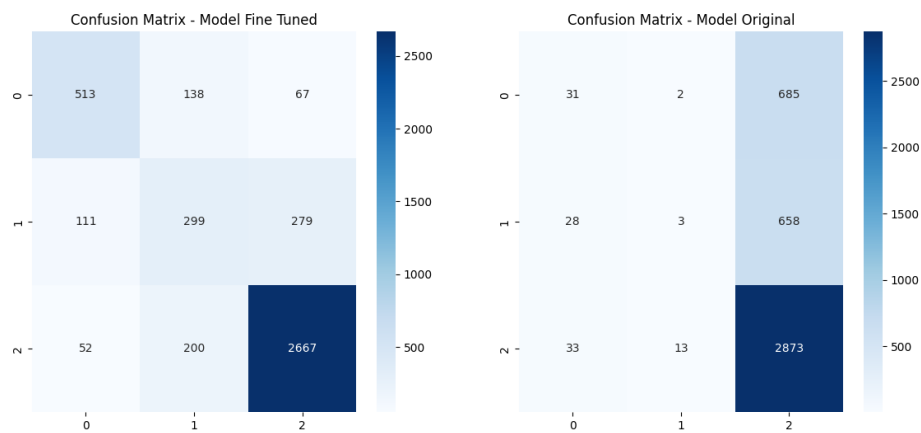


Fig. 5: Matrices de confusión modelo fine-tune y original

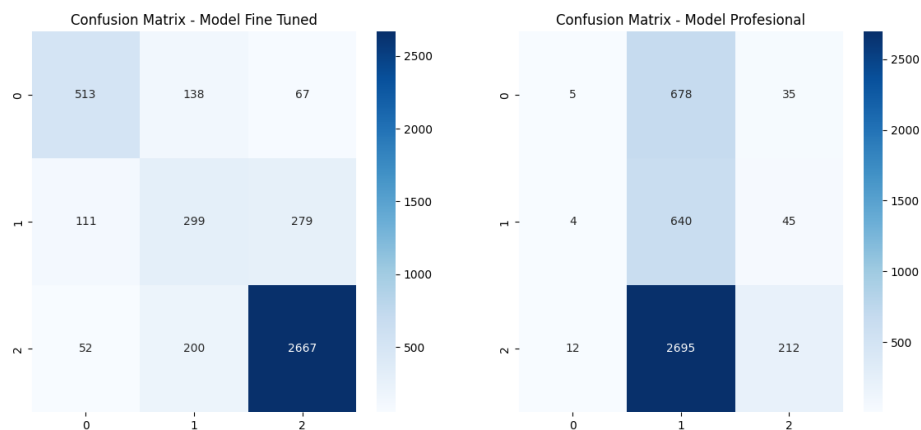


Fig. 6: Matrices de confusión modelo fine-tune y profesional

Table 7: Diferencias entre el modelo ajustado y el original

Métrica	Diferencia
Precisión	0.2783
Recall	0.6172
F1-Score	0.6501
Precisión Global	0.6172

El modelo ajustado supera al modelo original en todas las métricas, con una mejora especialmente notable en **F1-Score** y **precisión global**, lo que indica una mejor capacidad del modelo ajustado para clasificar correctamente las secuencias.

Finalmente, se realizó una comparación entre el modelo ajustado y el modelo profesional. Las diferencias observadas son las siguientes:

Table 8: Diferencias entre el modelo ajustado y el modelo profesional

Métrica	Diferencia
Precisión	0.2792
Recall	0.6200
F1-Score	0.6969
Precisión Global	0.6200

El modelo ajustado muestra una clara superioridad en términos de **precisión**, **recall**, y **F1-Score** frente al modelo profesional.

Los resultados muestran que el modelo ajustado mejora significativamente en todas las métricas en comparación con el modelo original y el modelo profesional.

Este proceso demuestra cómo modelos preentrenados como BERT pueden ser adaptados y mejorados para tareas específicas mediante el ajuste fino, siempre que se monitoreen cuidadosamente métricas de rendimiento para evitar problemas como el sobreajuste.

## 4 Sistema de Recomendaciones de Hoteles Basado en el Análisis de Emociones

La aplicación es un sistema web diseñado para gestionar los comentarios de los clientes sobre hoteles y el análisis de sentimientos basado en dichos comentarios. Utiliza el framework **Flask** <sup>[19]</sup> <sup>[25]</sup> para facilitar su desarrollo y funcionamiento. Además, incluye un sistema básico de administración que permite gestionar tanto los hoteles como los comentarios realizados por los usuarios.

La aplicación comienza cargando tres conjuntos de datos desde archivos CSV que contienen información relevante. El primer archivo, `users.csv`, almacena datos sobre los usuarios registrados, incluyendo su nombre, nombre de usuario y contraseña. El segundo archivo, `hotels.csv`, contiene información sobre los hoteles, como su nombre, dirección, imagen, descripción, puntuación total y cantidad de comentarios. Por último, `comments.csv` recoge los comentarios realizados por los usuarios sobre los hoteles junto con el análisis de sentimientos correspondiente.

El sistema cuenta con una funcionalidad de autenticación simple que permite a los usuarios iniciar sesión o registrarse. A través del endpoint `/login`, los usuarios pueden acceder a sus cuentas proporcionando su nombre de usuario y contraseña. Si las credenciales son válidas, se les redirige a la página principal de hoteles. Los nuevos usuarios pueden registrarse mediante el endpoint `/register`, donde deben proporcionar su nombre completo, nombre de usuario y contraseña.

Los administradores tienen acceso a un panel especial donde pueden gestionar todos los hoteles registrados en el sistema. Desde esta interfaz pueden agregar nuevos hoteles proporcionando toda la información necesaria como nombre, dirección e imagen.

Finalmente, tanto los usuarios como los administradores tienen la opción de cerrar sesión mediante una ruta específica que elimina su información actual y les redirige a la página principal.

La funcionalidad principal de la aplicación se centra en la gestión de comentarios sobre hoteles. En la página dedicada a los hoteles (`/hotels`), se presenta una lista completa de las opciones disponibles. Cada hotel tiene una página específica donde los usuarios pueden dejar comentarios y calificaciones. Estos comentarios son analizados utilizando el modelo de análisis de sentimientos nuestro, que determina si son positivos, negativos o neutrales. Este análisis ayuda a actualizar la puntuación total del hotel en función del feedback recibido.

Para organizar el ranking de los hoteles, se utilizó la salida del modelo junto con el rating. La salida del modelo consistió en una etiqueta (positivo, negativo

o neutro) y la confianza del modelo en la precisión de esa etiqueta. Dado que el valor de confianza refleja el grado de certeza del modelo respecto a la correctitud de la etiqueta devuelta, se interpretó como una medida de cuán cerca estaba la emoción expresada en una reseña con la etiqueta asignada.

Para calcular el valor correspondiente a cada reseña, se decidió sumar el valor de confianza proporcionado por el modelo y el rating asociado, ambos multiplicados por un factor que refleja la importancia otorgada a cada componente. En el caso del rating, se multiplicó por 1.5, mientras que la confianza se multiplicó por -0.5 si el sentimiento era negativo, por 1.0 si era neutro, y por 2.5 si era positivo. Posteriormente, para cada hotel, se realizó una sumatoria de todas las reseñas correspondientes y, finalmente, se ordenaron los resultados de menor a mayor.

El sistema actúa como un gestor integral para hoteles que permite a los usuarios dejar sus opiniones y recibir sugerencias basadas en análisis automatizados. Integra funcionalidades clave como autenticación segura y administración eficiente para garantizar una experiencia fluida tanto para usuarios como para administradores.



## 5 Resultados del Preprocesamiento de datos

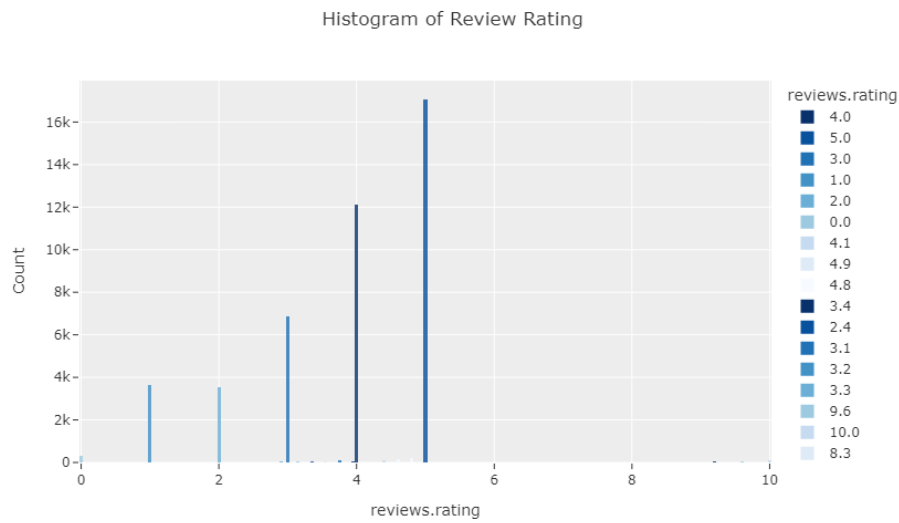


Fig. 7: Histograma para visualizar la distribución de las calificaciones de las reseñas en el conjunto de datos

Este análisis dió como resultado que el conjunto de entrenamiento presenta principalmente comentarios de 4 y 5 estrellas.

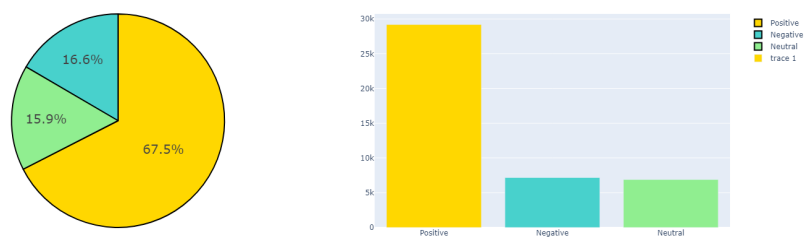


Fig. 8: La figura visualiza la distribución de etiquetas en el conjunto de datos. El gráfico circular a la izquierda muestra el porcentaje de cada etiqueta, mientras que el gráfico de barras a la derecha muestra el conteo de cada etiqueta.

Debido a que la mayor cantidad de reseñas tienen de 4 a 5 estrellas da como mayor porcentaje los positivos

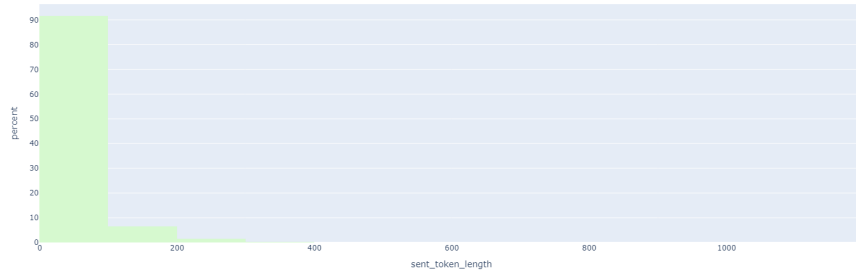


Fig. 9: Histograma para visualizar la distribución de las longitudes de los tokens de las oraciones en el conjunto de datos. Este gráfico representa la longitud de las oraciones (en tokens) en el eje x y el porcentaje de ocurrencias en el eje y. El histograma tiene 20 intervalos, agrupando los datos en rangos.

## 6 Aspectos a mejorar

- Aumentar el conjunto de datos recopilando más opiniones de hoteles para ayudar al modelo a dar resultados más precisos.
- Asegurar que las clases (positivas, neutras, negativas) estén equilibradas para evitar sesgos.
- Evaluar cómo utilizar la información brindada por los emojis y emoticonos en las opiniones ya que son eliminados antes de evaluar el sentimiento predominante en una reseña.
- Experimentar con diferentes configuraciones de hiperparámetros como la tasa de aprendizaje, tamaño del lote y número de épocas para buscar la más óptima.
- Implementar técnicas de regularización como Dropout <sup>[20]</sup> o Early Stopping <sup>[21]</sup> para prevenir el sobreajuste.
- Considerar el uso de modelos multilingües <sup>[24]</sup> si se decide expandir el sistema a otros idiomas.
- Ajustar el modelo principal o entrenar modelos adicionales en tareas específicas si se tiene acceso a opiniones etiquetadas (limpieza, servicio al cliente).
- Incorporar información adicional relevante para la valoración, como características del hotel y ubicación.
- Implementar un sistema para recopilar feedback sobre las predicciones del modelo y actualizar periódicamente con nuevos datos.

## References

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008). URL: <https://arxiv.org/abs/1706.03762>.
2. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186). URL: <https://arxiv.org/abs/1810.04805>.
3. Tunstall, L. (2019). Illustrated Guide to Transformers - Step by Step Explanation. Towards Data Science. URL: <https://towardsdatascience.com/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0>.
4. Google. (2024). Google Colab. URL: <https://colab.research.google.com/>.
5. Wikipedia. (2024). Inteligencia artificial. URL: [https://es.wikipedia.org/wiki/Inteligencia\\_artificial](https://es.wikipedia.org/wiki/Inteligencia_artificial).
6. Wikipedia. (2024). Procesamiento de lenguajes naturales. URL: [https://es.wikipedia.org/wiki/Procesamiento\\_de\\_lenguajes\\_naturales](https://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales).
7. Wikipedia. (2024). Análisis de sentimiento. URL: [https://es.wikipedia.org/wiki/An%C3%A1lisis\\_de\\_sentimiento](https://es.wikipedia.org/wiki/An%C3%A1lisis_de_sentimiento).
8. Wikipedia. (2024). Naive Bayes classifier. URL: [https://en.wikipedia.org/wiki/Naive\\_Bayes\\_classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier).
9. Wikipedia. (2024). Máquina de vectores de soporte. URL: [https://es.wikipedia.org/wiki/M%C3%A1quina\\_de\\_vectores\\_de\\_soporte](https://es.wikipedia.org/wiki/M%C3%A1quina_de_vectores_de_soporte).
10. Wikipedia. (2024). Aprendizaje profundo. URL: [https://es.wikipedia.org/wiki/Aprendizaje\\_profundo](https://es.wikipedia.org/wiki/Aprendizaje_profundo).
11. Wikipedia. (2024). Redes neuronales recurrentes. URL: [https://es.wikipedia.org/wiki/Redes\\_neuronales\\_recurrentes](https://es.wikipedia.org/wiki/Redes_neuronales_recurrentes).
12. testingbaires.com. (2024). Modelo de Proceso de Lenguaje Natural. URL: <https://testingbaires.com/modelo-de-proceso-de-lenguaje-natural/>.
13. Wikipedia. (2024). BERT (Bidirectional Encoder Representations from Transformers). URL: [https://es.wikipedia.org/wiki/BERT\\_\(modelo\\_de\\_lenguaje\)](https://es.wikipedia.org/wiki/BERT_(modelo_de_lenguaje)).
14. Wikipedia. (2024). Transformador generativo preentrenado. URL: [https://es.wikipedia.org/wiki/Transformador\\_generativo\\_preentrenado](https://es.wikipedia.org/wiki/Transformador_generativo_preentrenado).
15. IBM. (2024). ¿Qué es el análisis de sentimiento?. URL: <https://www.ibm.com/es-es/topics/sentiment-analysis>.
16. IBM. (2024). ¿Qué es la regresión logística?. URL: <https://www.ibm.com/es-es/topics/logistic-regression>.
17. Datafiniti. (2018). Hotel Reviews Dataset. URL: <https://data.world/datafiniti/hotel-reviews>.
18. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. URL: <https://aclanthology.org/N19-1423.pdf>.
19. Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media.
20. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958. URL: <https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf>.

21. Prechelt, L. (1998). Automatic early stopping using cross-validation: quantifying the criteria. *Neural Networks*, 11(4), 761-767. URL: <https://www.sciencedirect.com/science/article/pii/S0893608098001070>.
22. HuggingFace. (2021). nptown/bert-base-multilingual-uncased-sentiment. URL: <https://huggingface.co/nptown/bert-base-multilingual-uncased-sentiment>.
23. HuggingFace. (2021). bert-base-uncased. URL: <https://huggingface.co/google-bert/bert-base-uncased>.
24. HuggingFace. (2021). bert-base-multilingual-cased. URL: <https://huggingface.co/google-bert/bert-base-multilingual-cased>.
25. Flask. (2023). Flask Authentication and Authorization Documentation. URL: <https://flask.palletsprojects.com/en/2.3.x/security/>.