

Informe del Proyecto de Análisis de Sentimientos en textos largos

Joseph Yael Mendieta Avila - 506221037
Oscar Julián Ramirez Contreras - 506221006

Resumen—Este informe documenta el desarrollo de una aplicación web para el análisis de sentimientos utilizando *FastAPI* y el modelo preentrenado *nlptown/bert-base-multilingual-uncased-sentiment* de *Hugging Face Transformers*. La aplicación evalúa el sentimiento de texto ingresado y comentarios extraídos de Reddit, clasificándolos como positivos, negativos o neutrales. Este informe describe los errores encontrados, los logros obtenidos, los métodos utilizados para resolver problemas y los desafíos enfrentados durante el proceso de codificación.

I. INTRODUCCIÓN

Este informe documenta el desarrollo de una aplicación web para el análisis de sentimientos utilizando *FastAPI* y el modelo preentrenado *nlptown/bert-base-multilingual-uncased-sentiment* de *Hugging Face Transformers*. La aplicación evalúa el sentimiento de texto ingresado y comentarios extraídos de Reddit, clasificándolos como positivos, negativos o neutrales. Este informe describe los errores encontrados, los logros obtenidos, los métodos utilizados para resolver problemas y los desafíos enfrentados durante el proceso de codificación.

II. ERRORES ENCONTRADOS

Durante el desarrollo de la aplicación, nos enfrentamos a varios errores y problemas, los cuales se describen a continuación:

- **Error de Nombre No Definido:** Se encontró un error de nombre no definido para la función `filtro_publicacion`. Este error se resolvió definiendo correctamente la función.
- **Clasificación Desconocida:** Inicialmente, la clasificación del sentimiento a menudo se mostraba como "Desconocido". Esto se debió a un mapeo incorrecto de las clasificaciones devueltas por el modelo a sus correspondientes valores numéricos.
- **Filtro Incorrecto:** El filtro de publicación no funcionaba como se esperaba, permitiendo pasar comentarios con 1 o 2 estrellas. Esto se corrigió ajustando la lógica del filtro y los umbrales de clasificación.

III. LOGROS

A pesar de los desafíos, logramos varios hitos importantes:

- **Implementación del Modelo:** Integramos exitosamente el modelo de análisis de sentimientos de *Hugging Face Transformers* en la aplicación.
- **Filtro de Sentimiento Funcional:** Desarrollamos y ajustamos un filtro de sentimiento que clasifica correctamente los textos como positivos, negativos o neutrales.

- **Interfaz de Usuario:** Implementamos una interfaz de usuario simple utilizando *FastAPI* y archivos estáticos para la interacción del usuario.

IV. MÉTODOS UTILIZADOS

Para abordar los problemas y mejorar la funcionalidad de la aplicación, utilizamos los siguientes métodos:

- **Logger:** Implementamos un logger para rastrear y registrar las entradas y salidas, así como los errores durante la ejecución.
- **Debugging:** Utilizamos técnicas de depuración para identificar y corregir errores en el código.
- **Pruebas:** Realizamos pruebas exhaustivas con diferentes textos y comentarios para asegurar que la aplicación funcionara como se esperaba.

V. DESAFÍOS EN LA CODIFICACIÓN

El proceso de desarrollo presentó varios desafíos:

- **Integración del Modelo:** La integración del modelo de *Hugging Face Transformers* requirió comprender cómo utilizar el pipeline de análisis de sentimientos y mapear correctamente las clasificaciones.
- **Manejo de Errores:** Asegurar que la aplicación maneje correctamente los errores y proporcione mensajes de error informativos fue crucial para mejorar la experiencia del usuario.
- **Filtrado de Sentimientos:** Definir y ajustar los umbrales adecuados para el filtro de sentimientos fue un proceso iterativo que requirió pruebas y ajustes continuos.

VI. CÓDIGO FUENTE

A continuación se presenta el código fuente final de la aplicación:

```
1 import logging
2 from fastapi import FastAPI, HTTPException
3 from fastapi.responses import StreamingResponse
4 from starlette.staticfiles import StaticFiles
5 from transformers import pipeline
6 import requests
7
8 app = FastAPI()
9
10 # Configurar el logger
11 logging.basicConfig(level=logging.INFO)
12 logger = logging.getLogger(__name__)
13
14 clasificador = pipeline('sentiment-analysis', model=
15     'nlptown/bert-base-multilingual-uncased-
16     sentiment')
```

```

16 UMBRAL_POSITIVO = 3
17 UMBRAL_NEGATIVO = 2
18
19 app.mount("/static", StaticFiles(directory="static")
20         , name="static")
21
22 mapeo_clasificaciones = {
23     '5 stars': 5,
24     '4 stars': 4,
25     '3 stars': 3,
26     '2 stars': 2,
27     '1 star': 1
28 }
29
30 def filtro_publicacion(sentimiento: float) -> bool:
31     return sentimiento >= UMBRAL_POSITIVO
32
33 @app.get("/")
34 async def read_index():
35     return StreamingResponse(open("static/index.html", "rb"), media_type="text/html")
36
37 @app.get("/analisisTexto")
38 def analizar_sentimiento_con_filtro(texto: str):
39     logger.info(f"Texto recibido para analisis: {texto}")
40     resultado = clasificador(texto)
41     logger.info(f"Resultado del analisis de sentimientos: {resultado}")
42     clasificacion = resultado[0]['label']
43     sentimiento = mapeo_clasificaciones.get(
44         clasificacion, "Desconocido")
45     if sentimiento in [1, 2]:
46         sentimiento = 1
47     elif sentimiento == 3:
48         sentimiento = 2
49     else:
50         sentimiento = 3
51     cumple_filtro = filtro_publicacion(sentimiento)
52     mensaje = generar_mensaje_individual(sentimiento)
53
54     return {
55         "texto": texto,
56         "clasificacion": clasificacion,
57         "sentimiento": sentimiento,
58         "cumple_filtro": cumple_filtro,
59         "mensaje": mensaje
60     }
61
62 @app.get("/analizarComentariosReddit")
63 def analizar_comentarios_reddit_con_filtro(subreddit: str, limit: int = 10):
64     logger.info(f"Subreddit recibido para analisis: {subreddit}")
65     url = f"https://www.reddit.com/r/{subreddit}/comments.json?limit={limit}"
66     headers = {'User-agent': 'sentiment-analyzer'}
67     response = requests.get(url, headers=headers)
68
69     if response.status_code != 200:
70         raise HTTPException(status_code=500, detail="Error al obtener los comentarios de Reddit")
71
72     comentarios = response.json()['data']['children']
73     resultados_comentarios = []
74
75     for comentario in comentarios:
76         texto = comentario['data']['body']
77         resultado = clasificador(texto)
78         logger.info(f"Comentario: {texto}, Resultado del analisis de sentimientos: {resultado}")
79         clasificacion = resultado[0]['label']
80         sentimiento = mapeo_clasificaciones.get(
81             clasificacion, "Desconocido")
82
83         if sentimiento in [1, 2]:
84             sentimiento = 1
85         elif sentimiento == 3:
86             sentimiento = 2
87         else:
88             sentimiento = 3
89         cumple_filtro = filtro_publicacion(sentimiento)
90         mensaje = generar_mensaje_individual(sentimiento)
91
92         resultados_comentarios.append({
93             "texto": texto,
94             "clasificacion": clasificacion,
95             "sentimiento": sentimiento,
96             "cumple_filtro": cumple_filtro,
97             "mensaje": mensaje
98         })
99
100     mensaje = generar_mensaje_reddit(
101         resultados_comentarios)
102     return {
103         "subreddit": subreddit,
104         "comentarios_analizados":
105             resultados_comentarios,
106         "mensaje": mensaje
107     }
108
109 def generar_mensaje_reddit(comentarios: list) -> str:
110     if not comentarios:
111         return "No se encontraron comentarios que cumplan con los criterios de filtro."
112
113     comentarios_filtrados = [comentario for comentario in comentarios if comentario['cumple_filtro']]
114     if not comentarios_filtrados:
115         return "No hay comentarios que pasen el filtro."
116
117     clasificaciones_numericas = [comentario['sentimiento'] for comentario in comentarios_filtrados]
118     clasificacion_promedio = sum(
119         clasificaciones_numericas) / len(
120         clasificaciones_numericas)
121
122     if clasificacion_promedio >= 4.0:
123         mensaje = "Los comentarios de este tema en Reddit son muy positivos!"
124     elif 2.0 <= clasificacion_promedio < 4.0:
125         mensaje = "Los comentarios de este tema en Reddit reflejan opiniones mixtas."
126     else:
127         mensaje = "Los comentarios de este tema en Reddit son principalmente negativos o criticos."
128     return mensaje
129
130 def generar_mensaje_individual(sentimiento: float) -> str:
131     if sentimiento >= 3:
132         mensaje = "El analisis de sentimientos indica una gran positividad en el texto."
133     elif sentimiento == 2:
134         mensaje = "El analisis de sentimientos revela una neutralidad en el texto."
135     else:
136         mensaje = "El analisis de sentimientos detecta un tono negativo en el texto."
137     return mensaje

```

VII. CONCLUSIÓN

- Además, este proyecto nos lleva a reflexionar sobre la importancia del análisis de sentimientos en el mundo digital actual. No solo es un ejercicio técnico, sino también una oportunidad para comprender y cuantificar las emociones en línea. En un mundo donde la comunicación en línea es omnipresente, las herramientas que pueden analizar y comprender los sentimientos expresados en texto son cada vez más relevantes.
- El análisis de sentimientos tiene el potencial de impactar significativamente una amplia gama de sectores, desde el marketing y la publicidad hasta la atención al cliente y la salud mental. Al proporcionar una forma rápida y automatizada de evaluar el tono y el sentimiento de grandes volúmenes de texto, las herramientas de análisis de sentimientos pueden ayudar a las empresas a comprender mejor las necesidades y preferencias de sus clientes, identificar tendencias emergentes en el mercado y mejorar la calidad de sus productos y servicios.
- Si bien hemos logrado desarrollar una aplicación funcional de análisis de sentimientos, reconocemos que todavía hay desafíos por delante. La mejora continua de la precisión y la eficiencia de los modelos de procesamiento de lenguaje natural sigue siendo un área de investigación activa y desarrollo. Además, la ética y la privacidad deben ser consideraciones fundamentales en el desarrollo y la implementación de estas herramientas, especialmente cuando se trata de analizar datos sensibles o personales. Mirando hacia el futuro, vemos oportunidades emocionantes para colaborar con expertos en diversos campos y aplicar el análisis de sentimientos para abordar problemas y desafíos sociales en constante evolución.

REFERENCIAS

- FastAPI: <https://fastapi.tiangolo.com/>
- Modelo preentrenado de Hugging Face Transformers: <https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment>
- Cuaderno de Google Colab: <https://colab.research.google.com/drive/1AJ3s7KGfmmYKNIqSOqRWH7DrrdK4YI3c#scrollTo=NZR-bLaEY-qP>