



Proyecto final NLP: Uso de APIs e Interfaces gráficas

Inteligencia Artificial Avanzada Para la Ciencia de Datos II

Eduardo Martinez Martinez – A01023975

Equipo 4

03.12.2023

El objetivo principal de este proyecto es el desarrollo de una interfaz de usuario interactiva y avanzada que integre tecnologías de Procesamiento de Lenguaje Natural (NLP) y Reconocimiento de Voz. Esta interfaz busca proporcionar una forma accesible y eficiente para que los usuarios interactúen con sistemas de inteligencia artificial complejos, transformando el modo en que interactuamos con la tecnología en nuestro día a día.

Este enfoque implica un desafío significativo: facilitar la interacción con las APIs de Lenguaje Natural de manera que los usuarios puedan realizar tareas como análisis de sentimientos, clasificación de texto y traducción de manera intuitiva. El proyecto no solo pone a prueba la habilidad técnica para integrar estas tecnologías avanzadas en una interfaz de usuario, sino que también desafía nuestra capacidad para hacer estas herramientas comprensibles y útiles para un público más amplio.

Además, la incorporación del Reconocimiento de Voz en la interfaz es un paso crucial hacia la creación de una experiencia de usuario más natural y accesible. Esto permite a los usuarios interactuar con la aplicación utilizando comandos de voz, lo cual no solo es una comodidad, sino también una herramienta esencial para hacer la tecnología accesible a usuarios con diferentes capacidades y preferencias.

En conjunto, el proyecto busca ir más allá de la mera funcionalidad técnica; se trata de diseñar una experiencia de usuario que sea intuitiva, eficiente y, sobre todo, accesible. Este enfoque refleja una comprensión profunda de cómo las tecnologías de NLP y Reconocimiento de Voz pueden ser aplicadas de manera práctica y significativa en la vida cotidiana, subrayando el potencial transformador de la inteligencia artificial en la interacción humana y la comunicación.

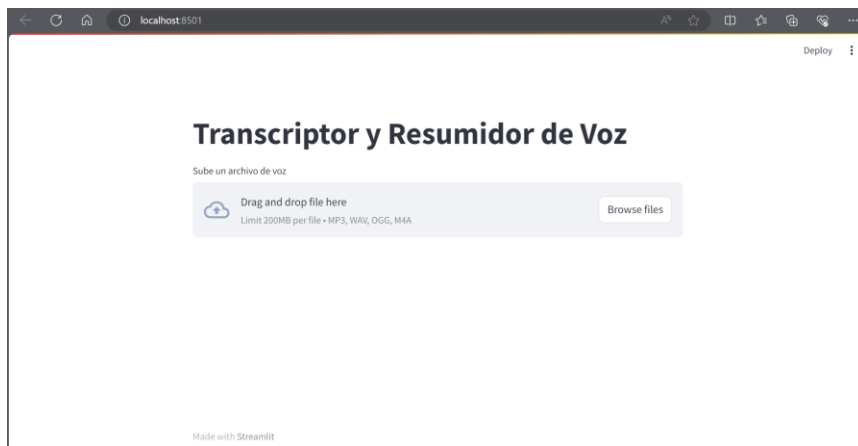
## Comentarios Personales:

Desde una perspectiva personal, este proyecto representa una oportunidad fascinante para explorar el cruce entre la tecnología de inteligencia artificial y la interacción humana. La interfaz actúa como un puente entre la complejidad técnica de las APIs de NLP y la simplicidad que los usuarios finales desean. Es un reto que no solo demanda habilidades técnicas en programación y diseño de sistemas, sino también una comprensión profunda de la experiencia del usuario.

Además, el proyecto pone de relieve la importancia de la inteligencia artificial en la vida cotidiana y su potencial para transformar cómo interactuamos con la tecnología. La capacidad de analizar y responder a la voz y al texto en lenguaje natural abre nuevas puertas para la accesibilidad y la eficiencia en diversas aplicaciones, desde asistencia personal hasta análisis de datos complejos.

En resumen, el proyecto es una combinación emocionante de desafíos técnicos y creativos, con el potencial de avanzar significativamente en cómo interactuamos con las máquinas en nuestra vida diaria.

Interfaz antes de usarse:



Transcripción:



Resumen:



## Código:

```
import streamlit as st
1 from openai import OpenAI
2
3 # Configuración del cliente OpenAI
4 client = OpenAI()
5
6 def transcribe_audio(audio_file):
7     response = client.audio.transcriptions.create(
8         model="whisper-1",
9         file=audio_file
10    )
11    # Inspecciona la respuesta para entender su estructura
12    print(response)
13
14    # Suponiendo que la respuesta tiene un atributo 'text', cambia la siguiente línea
15    return response.text # Cambia 'text' por el atributo correcto según la inspección
16
17
18 def summarize_text(text):
19     response = client.chat.completions.create(
20         model="gpt-4-1106-preview",
21         messages=[
22             {"role": "system", "content": "You are a helpful assistant."},
23             {"role": "user", "content": text}
24         ]
25     )
26     # Inspecciona la respuesta para entender su estructura
27     print(response)
28
29     # Modifica la siguiente línea según la estructura correcta de la respuesta
30     # Por ejemplo, si la respuesta es un objeto con un atributo 'content', usa
31     response.content
32     return response.choices[0].message.content # Modifica esta línea según la estructura de
33 la respuesta
34
35
36 # Diseño de la interfaz de Streamlit
37 st.title('Transcriptor y Resumidor de Voz')
38
39 uploaded_file = st.file_uploader("Sube un archivo de voz", type=['mp3', 'wav', 'ogg', 'm4a'])
40 if uploaded_file is not None:
41     # Transcripción del archivo
42     text = transcribe_audio(uploaded_file)
43     st.text_area("Transcripción", text, height=250)
44
45     # Generación y visualización del resumen
46     summary = summarize_text("Haz un resumen de lo siguiente: " + text)
47     st.text_area("Resumen", summary, height=150)
```