

Evidencia NLP | Speech Recognition and NLP

Diego Rodríguez A00829925

24 de noviembre de 2023

1. Descripción del proyecto

Este ejercicio práctico está diseñado para cultivar una variedad de habilidades esenciales en el campo de la inteligencia artificial y el desarrollo de aplicaciones web. Primero, se enfoca en el uso de modelos de reconocimiento automático de voz (ASR) preentrenados, como Whisper, que son fundamentales para convertir archivos de audio en texto transcrito. Esto permite que los datos hablados sean analizados y procesados por algoritmos de procesamiento del lenguaje natural (NLP).

En segundo lugar, el ejercicio destaca la aplicación de modelos avanzados de NLP para sintetizar y resumir el texto, proporcionando un enfoque práctico para la comprensión de cómo las tecnologías de IA pueden extraer y condensar la información más relevante de grandes volúmenes de datos textuales. Además, este proyecto incluye la integración con la API de OpenAI, lo que demuestra cómo se puede conectar un modelo de NLP a una aplicación en desarrollo. Esto es esencial para cualquier desarrollador que busque incorporar capacidades de inteligencia artificial en aplicaciones de software.

Por último, el uso de Streamlit como marco para la creación de una aplicación web interactiva subraya una competencia crucial en la visualización de datos y la creación de interfaces de usuario intuitivas. Streamlit permite a los desarrolladores desplegar rápidamente aplicaciones con capacidades de IA, lo que es crucial para la demostración y el uso práctico de los modelos entrenados.

En conjunto, este ejercicio proporciona una experiencia de aprendizaje integral que abarca desde el reconocimiento de voz hasta el resumen de texto utilizando tecnologías de IA de vanguardia, todo dentro de una aplicación web accesible.

2. Demostración de la webapp

A continuación se presentaran capturas de pantalla de la webapp creada en streamlit para el ejercicio. En la primera imagen se muestra el título de la aplicación, así como una barra reproductora del audio a trabajar. Debajo de ella se redacta el transcript del audio hecho por el algoritmo de Speech Recognition Whisper, hosteado por la librería OpenAI. En la segunda imagen se muestra el final de la webapp donde se enlista un resumen con los puntos

más importantes del texto obtenido, esto mediante la API de OpenAI para hacer uso de su algoritmo de inteligencia artificial generativa GPT-3.5 Turbo.

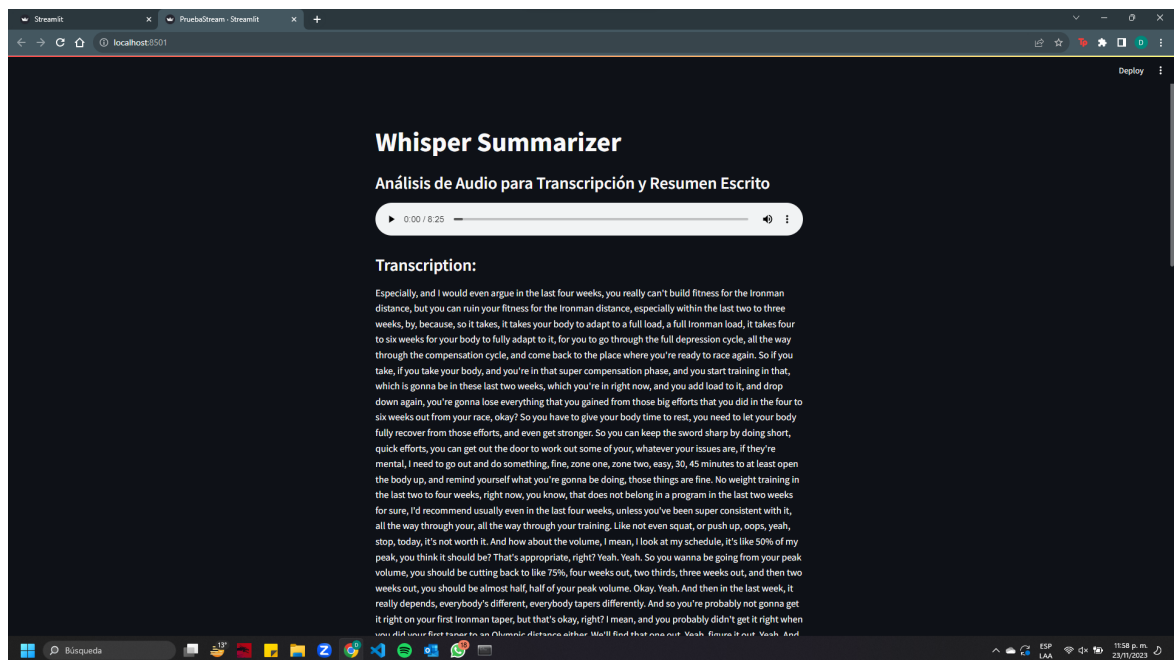


Figura 1: Reproductor de Audio y Transcript hecho con Speech Recognition Whisper

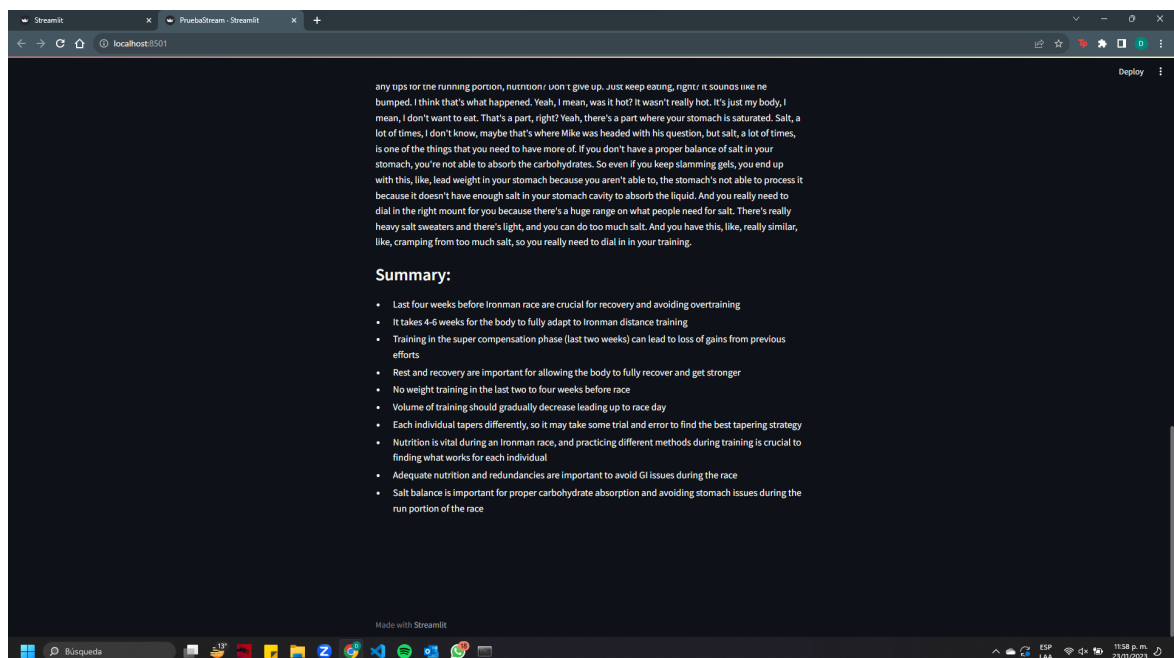


Figura 2: Summary del Transcript procesado con API de OpenAI

3. Código de la webapp

```
import streamlit as st
import openai
import ssl
import certifi

ssl._create_default_https_context = lambda: ssl.
    create_default_context(cafile=certifi.where())
print(ssl.get_default_verify_paths())

st.title("Whisper Summarizer")
st.subheader("Análisis de Audio para Transcripción y
    Resumen Escrito")

OPENAI_API_KEY = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    "
client = openai.OpenAI(api_key=OPENAI_API_KEY)

def transcribe_audio(file_path):
    with open("MA1.m4a", "rb") as audio_file:
        transcript = client.audio.transcriptions.create(
            model="whisper-1",
            file=audio_file
        )
    return transcript.text

def custom_chatgpt(user_input):
    messages = [
        {"role": "system", "content": "You are an office
            administrator, summarize the text in key points
            "},
        {"role": "user", "content": user_input}
    ]
    response = client.chat.completions.create(
        model="gpt-3.5-turbo",
        messages=messages
    )
```

```
        chatgpt_reply = response.choices[0].message.content
    return chatgpt_reply

def main():
    file_path = 'MA1.m4a'
    st.audio(file_path, format="audio")

    # Transcribe audio
    st.subheader("Transcription:")
    transcription = transcribe_audio(file_path)
    st.write(transcription)

    # Summarize using ChatGPT
    st.subheader("Summary:")
    summary = custom_chatgpt(transcription)
    st.write(summary)

if __name__ == "__main__":
    main()
```