



**Tecnológico
de Monterrey**

Instituto Tecnológico y de Estudios Superiores de Monterrey

Escuela de Ingeniería y Ciencias

Inteligencia artificial avanzada para la ciencia de datos II

Ingeniería en Ciencias de Datos y Matemáticas

Módulo 3: Procesamiento de Lenguaje Natural (NLP)

Profesor

Dr. Juan Arturo Nolasco Flores

Federico Medina García Corral

A01721441

Monterrey, Nuevo León. 23 de noviembre de 2023

1. Descripción del Proceso

El código que se creó es una aplicación de Streamlit simple. Este tiene como objetivo transcribir un audio que se le proporciona a la aplicación y finalmente te muestra el texto del audio al igual que un resumen en bullet points agarrando las ideas principales de este. Para que funcione la aplicación, se utilizó la API de Whisper de OpenAI para poder transcribir el archivo de audio que se le proporciona para después utilizar el API de ChatGPT y hacer el resumen en bullet points de las ideas principales. Como se puede observar en la Figura 1, el código las variables `openai.api_key` y `model = whisper.load_model('base')` representan tanto la API de ChatGPT como el modelo de Whisper. Después se tienen dos funciones que son de suma importancia para la ejecución del código:

- `transcribe_audio(model, file_path)`: Esta función utiliza el modelo de Whisper para hacer el transcript del audio ubicado en el `file_path` que se asigna.
- `Custom_ChatGPT(user_input)`: Esta función se encarga de hacer el resumen del texto que se le envía mediante la variable `user_input`.

Finalmente, vemos como hay varias variables con la variable `st`, estas únicamente representan la interfaz que se hace mediante Streamlit para poner títulos, subtítulos y el texto que le añadirá el usuario, el cual es el path al audio. Finalmente tenemos `if file_path` el cual representa que si se puso un path al textbox, se correrá la función `transcribe_audio(model, file_path)` sobre ese audio y finalmente la función `Custom_ChatGPT(user_input)` sobre el transcript que se haya obtenido anteriormente.

2. Evidencias del Funcionamiento del Programa

```
1 import streamlit as st
2 import openai
3 #https://medium.com/gitconnected/using-the-whisper-api-to-transcribe-audio-files-45fb36d1aa1b
4
5 import whisper
6 # brew install ffmpeg # you may need to install
7
8
9 openai.api_key = 'COLOCAR API DE CHATGPT (LO BORRÉ PARA QUE NO ME COBREN EN CASO DE QUE ALGUIEN AGARRE EL CÓDIGO)'
10 model = whisper.load_model("base")
11
12 # Función para transcribir audio
13 def transcribe_audio(model, file_path):
14     transcript = model.transcribe(file_path)
15     return transcript['text']
16
17 # Función para la interacción con ChatGPT
18 def Custom_ChatGPT(user_input):
19     messages = [
20         {"role": "system", "content": "You are an office administrator, summarize the text in key points"},
21         {"role": "user", "content": user_input}
22     ]
23     response = openai.ChatCompletion.create(
24         model="gpt-3.5-turbo",
25         messages=messages
26     )
27     ChatGPT_reply = response["choices"][0]["message"]["content"]
28     return ChatGPT_reply
29
30
31 st.title("Transcripción y Resumen de Audio")
32
33 # Subida del archivo de audio
34 file_path = st.text_input("Enter the path to the audio file (Ej: /Users/federicom Medina/Documents/Repositorios/Whisper-ChatGPT-Audio/MA1.m4a)")
35
36 if file_path:
37     # Transcripción de audio
38     st.subheader("Transcripción:")
39     transcription = transcribe_audio(model, file_path)
40     st.write(transcription)
41
42     # Resumen con ChatGPT
43     st.subheader("Resumen:")
44     summary = Custom_ChatGPT(transcription)
45     st.write(summary)
```

Figura 1: Código Completo de la Actividad

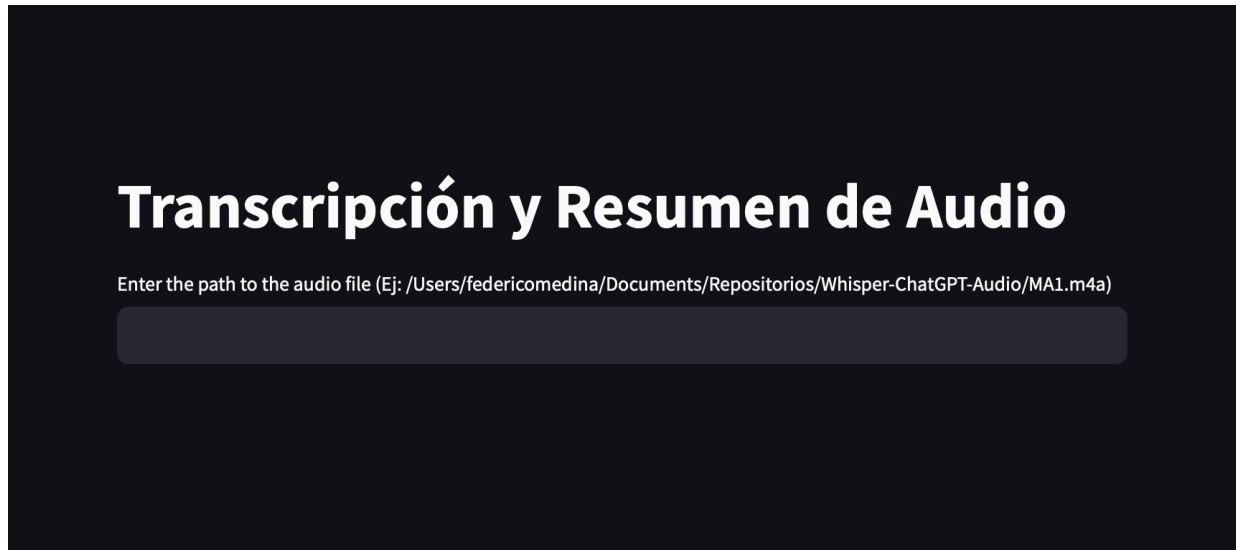


Figura 2: Figura que representa cómo el usuario ve la interfaz al ingresar

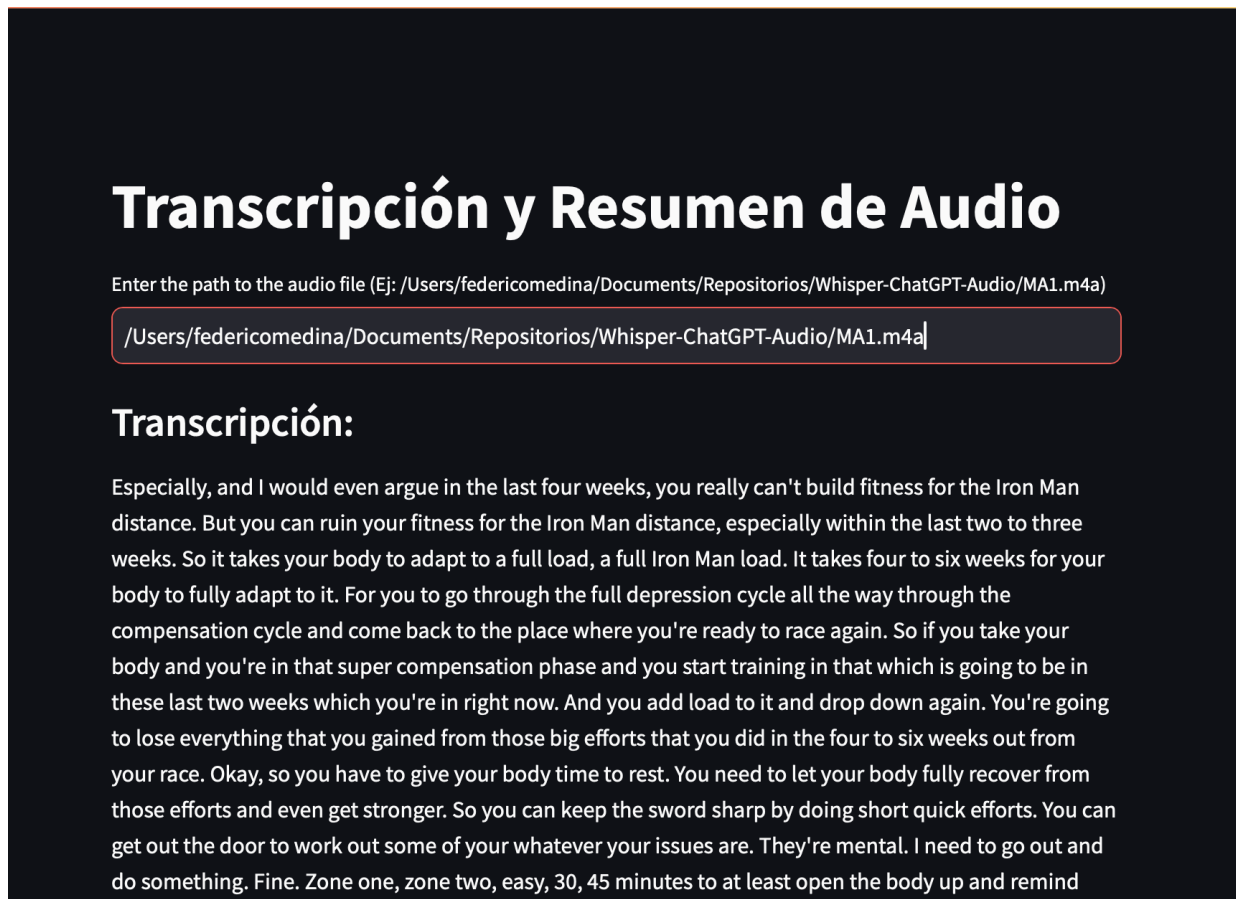


Figura 3: Parte de cómo se ve la interfaz al ingresar un path correcto (Transcript)

to absorb the carbohydrates. So, even if you keep slamming gels, you're, you're, you're, you end up with this like lead weight in your stomach because you aren't able to, the stomach's not able to process it because it doesn't have enough salt in your stomach cavity to absorb the liquid. And you really need dialent, the right amount for you, because there's a huge range on what people need for salt. There's really heavy salt sweaters and there's light, and you can do too much salt. And you have this like really similar light cramping from too much salt, so you really need a dial in in your training.

Resumen:

- Building fitness for an Iron Man distance is not possible in the last four weeks leading up to the race
- It takes four to six weeks for the body to fully adapt to the Iron Man load
- Training in the last two weeks can ruin the progress made in the previous four to six weeks
- Rest and recovery are important during this phase, with light workouts to keep the body active
- Weight training should be avoided in the last two to four weeks unless consistently incorporated throughout the training period
- Volume should be gradually reduced in the weeks leading up to the race, with about 50% of peak volume two weeks out
- Tapering strategies may vary for each individual and may require trial and error to find the most effective approach
- Adequate nutrition is crucial for Iron Man performance, and it is recommended to practice different strategies during training to find what works best
- Redundancies should be built into nutrition plans to account for potential mishaps during the race
- Salt intake is important for proper carbohydrate absorption and cramping can occur due to imbalances in salt levels
- Finding the right amount of salt for individual needs is crucial and can be determined through training trials

Figura 4: Parte de cómo se ve la interfaz al ingresar un path correcto (Summary)