

Reporte de Portafolio

Modulo 3 - Natural Language Processing Jorge Eduardo de León Reyna - A00829759

Este proyecto en Python utiliza las bibliotecas *whisper* para la transcripción de audio y *streamlit* para la interfaz de usuario. La aplicación permite cargar un archivo de audio y transcribir su contenido a texto, luego utiliza la API de *OpenAI* para generar un resumen del texto transcrito.

El proceso comienza solicitando al usuario que ingrese la ruta del archivo de audio. Posteriormente, al presionar el botón "Predict Sentiment", se realiza la transcripción del audio utilizando la función *transcribe_audio* que utiliza el modelo de lenguaje Whisper. Luego, el texto transcrito se pasa a la función *CustomChatGPT*, que utiliza la API de OpenAI para generar un resumen del texto en puntos clave.

El resumen generado se muestra en la interfaz de usuario mediante *Streamlit*. El modelo GPT-3.5-turbo de OpenAI se utiliza para aprovechar sus capacidades de procesamiento del lenguaje natural y generar resúmenes significativos. El código también muestra mensajes informativos en la interfaz de usuario, indicando el progreso y las acciones realizadas.

En conclusión, este proyecto integra varias tecnologías y servicios para lograr la transcripción de audio y la generación de resúmenes de texto de manera eficiente y amigable para el usuario, mostrando el potencial de las bibliotecas y servicios disponibles en el campo de Procesamiento del Lenguaje Natural.

Código

```
Python
import openai
import whisper
import streamlit as st

openai.api_key = 'sk-mRkkSBsvIhVEH2vXjDBBT3B1bkFJWjbgxE0IpR6ojupVv5iK'
model = whisper.load_model("base")

st.title("Audio summarize using whisper")
st.subheader("Portfolio - NLP Module")
st.subheader("Jorge de León - A00829759")
st.write('\n\n')

file_path = st.text_input("Enter a the audio file path to analyze", "Write Here...")

def transcribe_audio(model, file_path):
    transcript = model.transcribe(file_path)
    return transcript['text']
```

```

def CustomChatGPT(user_input):
    messages = [{"role": "system", "content": "You are an office administer, summarize the text in key points"}]
    messages.append({"role": "user", "content": user_input})
    response = openai.ChatCompletion.create(
        model = "gpt-3.5-turbo",
        messages = messages
    )
    ChatGPT_reply = response["choices"][0]["message"]["content"]
    return ChatGPT_reply

if st.button('Predict Sentiment'):
    st.write("Generating summarized text..")
    transcription = transcribe_audio(model, file_path)
    summary = CustomChatGPT(transcription)
    print(summary)
    st.subheader("Summarized text")
    st.write(summary)
else:
    st.write("Press the above button..")

```

Interfaz de Usuario

