TEXTO-VOZ

1. Introducción:

La herramienta de IA que se va a utilizar para este proyecto es whisper OpenAI la cual que simplifica enormemente el proceso de transcripción de audio a texto. Gracias a esta tecnología, convertir grabaciones de entrevistas, reuniones o cualquier otro tipo de audio en texto escrito es ahora más rápido y preciso que nunca.

Beneficios clave de Whisper OpenAI:

Facilidad de uso: La interfaz de Whisper OpenAI está diseñada para ser intuitiva, lo que permite a usuarios de todos los niveles comenzar a transcribir sus archivos de audio en cuestión de minutos.

Precisión: Los algoritmos avanzados de Whisper OpenAI ofrecen una alta tasa de precisión en la transcripción, lo que significa que podrás confiar en la exactitud del texto generado.

Versatilidad: Esta herramienta puede ser utilizada por una amplia gama de profesionales, desde periodistas e investigadores hasta estudiantes y cualquier persona que necesite convertir audio en texto.

https://neuroflash.com/es/whisper-openai-convertir-voz-en-texto-como-un-profesional/

2. Codigo y explicación:

#Importamos las librerias necesarias

import os

import streamlit as st

import whisper

from langchain.chains import LLMChain

from langchain.llms import OpenAI

from langchain.prompts import PromptTemplate

from langdetect import detect  # Librería para detectar el idioma

# Para trabajar con OpenAi debemos tener una ApiKey, la cual almacenamos en el equipo

#aqui le indicamos donde esta almacenada

with open('C:/ProyectoChatOpenAI/Scripts/Api\_key.txt') as f:

    api\_key = f.read().strip()

# Procedemos ahora a configurar la clave de API de OpenAI en LangChain

os.environ["OPENAI\_API\_KEY"] = api\_key

# Aqui cargamos el modelo pre-entrenado de Whisper para la transcripción

model = whisper.load\_model("base")

# Configuración de la aplicación en Streamlit, para todo el frontend de la aplicación

st.title("Aplicación de Transcripción de Audio y Análisis de Lenguaje")

st.write("Esta aplicación transcribe audio y genera un texto usando LangChain.")

# Desde la aplicación solicitamos la subida de archivo de audio al usuario

audio\_file = st.file\_uploader("Sube un archivo de audio para transcribir", type=["mp3", "wav"])

if audio\_file:

    # Primero vamos a guardar el archivo de audio temporalmente

    with open("temp\_audio.mp3", "wb") as f:

        f.write(audio\_file.getbuffer())

    # Ahora ya vamos a transcribir el audio usando Whisper

    result = model.transcribe("temp\_audio.mp3")

    transcription\_text = result["text"]

    st.write("Transcripción del Audio (antes de traducir, si es necesario):")

    st.write(transcription\_text)

    # En estos pasos debido a nuestra ubicacion vamos a validar que el idiona sea español, primero vamos a detectar el idioma del texto transcrito.

    detected\_language = detect(transcription\_text)

    # Despues de hacer la validación si el texto no está en español, debe proceder a traducirlo e informar al usuario

    if detected\_language != 'es':

        st.write("El texto no está en español, se procederá a traducirlo.")

    # Aqui se procede a solicitar traducción al español

        prompt\_template\_translation = PromptTemplate(

            input\_variables=["text"],

            template="Traduce el siguiente texto al español: {text}"

        )

    # Aqui configuramos el modelo de lenguaje con OpenAI

        llm\_translation = OpenAI(

            model\_name="gpt-3.5-turbo",  # Usamos este modelo

            temperature=0.5,             #Dejamos una temperatrura baja para que no alucine

            openai\_api\_key=api\_key  # Aqui llamamos la api key

        )

        chain\_translation = LLMChain(llm=llm\_translation, prompt=prompt\_template\_translation)

        # Si es necesario trabduce el texto a español e informa al usuario

        translated\_text = chain\_translation.run(text=transcription\_text)

        st.write("Texto traducido al español:")

        st.write(translated\_text)

        # Aqui le indicamos que actualize la transcripción con el texto traducido e informe al usuario

        transcription\_text = translated\_text

    else:

        st.write("El texto ya está en español.")

    # Para complementar el proyecto vamos a crear un modelo de lenguaje para el análisis ya en español

    prompt\_template\_summary = PromptTemplate(

        input\_variables=["text"],

        template="Haz un resumen del siguiente texto en español: {text}"  # Aqui validamos que el resumen esté en español

    )

    # Configuramos el modelo de lenguaje para el resumen

    llm\_summary = OpenAI(

        model\_name="gpt-3.5-turbo",  # Nuevamente definimos el modelo igual al anterior.

        temperature=0.5, #Dejamos una temperatura cercana a cero para que no alucine

        openai\_api\_key=api\_key #Cargamos nuesta api key

    )

    chain\_summary = LLMChain(llm=llm\_summary, prompt=prompt\_template\_summary)

    # Finalmente se genera el texto y se entrega en pantalla al usuario

    summary = chain\_summary.run(text=transcription\_text)

    st.write("Resumen Generado:")

    st.write(summary)

3. Ejecucion de la aplicación, nos ubicamos en la ruta donde esta alojado el codigo y desde cmd ejecutamos la aplicación:

Texto

Descripción generada automáticamente

La cual se carga de manera automatica en el equipo local con el puerto TCP 8502, para verlo desde otro equipo debemos usar la direccion Ip.

En la siguiente imagen vemos la carga de la aplicación y su ventana principal, en la cual se indica al usuario que cargue un archivo de extension .mp3 o wav con un maximo de 200 Mb

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

4. Aquí vamos a cargar el audio de un video de educación referente a la enseñanza de logica de programacion.

Captura de pantalla de un celular

Descripción generada automáticamente

5. Aquí carga nuestro archivo Lógica.mp3, el cual pesa 12.7 Mb

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

6. Aquí ya inicia el proceso el cual dura varios minutos.

Texto

Descripción generada automáticamente

7. Finalmente tenemos el trexto del archivo MP3 entregado



8. Este es el resultado completo del ejercicio, junto al codigo se adjunta el .mp3.

Resultado:

Transcripción del Audio (antes de traducir, si es necesario):

¿Quieres dominar la programación y dar vida a tus ideas? Entonces, este video es para ti. Aquí aprenderás los secretos de la lógica de programación desde cero, descubrirás cómo programar paso a paso y después de ver este video para ti va a ser increíblemente fácil aprender cualquier lenguaje de programación. Porque estos conceptos se aplican a todos los lenguajes de programación. Todo esto lo aprenderás en 10 minutos y puedes empezar a practicar y crear proyectos en una semana. Esto es el ingeniero de software programador X y si quieres aprender más sobre informática, computación y desarrollo web te invito a que te suscribas. Así que antes de aprender todos estos conceptos veamos para qué sirve la programación. Primero, la programación sirve para manipular y analizar datos. Por ejemplo, puedes transformar texto de minúsculas a mayúsculas. También puedes hacer operaciones como sumar números, incluso puedes enviar datos de un computador a otro. Así es como estos videos llegan a ti. Segundo, con la programación puedes dar instrucciones de hardware. Por ejemplo, puedo dar clic en un botón y prender mi computador. Existe código y mi cámara que permiten focarse en mi rostro. Y todas estas luces que ves en el teclado y el computador funcionan detrás de cena con código. Y tercero, la programación te permite crear páginas web, aplicaciones y herramientas de automatización. Por ejemplo, los juegos que juegas en Steam están creados con programación. Las redes sociales como TikTok y Facebook están creados con programación. Las herramientas de inteligencia artificial como chatGPT son creadas por programadores. Y los programadores más avanzados han creado hasta sistemas bancarios con la programación. Como, por ejemplo, PayPal. Yo el día de hoy como ingeniero de software estoy trabajando en un proyecto inmenso de programación, detectando terremotos en la ciudad donde vivo. Y con esto, te puedes dar cuenta que la programación está en todos lados. Ahora, el lenguaje de programación que aprendas, si importa. Por ejemplo, si quieres crear páginas web, aprendes JavaScript. Porque JavaScript es el lenguaje de la web. Si quieres trabajar con datos, machine learning o inteligencia artificial, utilizarías un lenguaje como Python, porque tiene bastantes herramientas para trabajar con datos. Ahora que ya entiendes la programación, veamos todos los conceptos claves de la lógica de programación. Y te dejo un e-book bajo el video para que puedas seguir todos estos conceptos mientras ves el video. Y hago esto porque realmente deseo que tengas una excelente experiencia aprendiendo a programar. Para aprender a programar desde cero, necesitas conocer sobre variables y tipos de datos. Y no te preocupes que todo esto lo voy a hacer muy fácil. Los datos son simplemente información del mundo. Y los tipos de datos son la manera en que un computador entiende esta información. Por ejemplo, puede ser un nombre como el de mi canal, programador X, puede ser un número como el número de vistas de este video, o puede ser una lista como la lista de todos los videos que ves en YouTube. Y ahora, ¿cómo guardamos estos datos en un computador? Utilizando las variables. Las variables son simplemente nombres que nosotros escribimos para guardar estos datos en el computador. Así que ahora guardemos estos tipos de datos en variables. Y empecemos con los tipos de datos primitivos. Por ejemplo, aquí estoy guardando el título de un libro que es un texto en una variable que se llama libro. Aquí guardamos un dato de número en una variable que se llama número, como tipos de datos tienes números enteros y también decimales. Por ejemplo, un número puede ser tres y un decimal puede ser 2.3. Ahora también tenemos un tipo de dato que se llama buleano. Y este dato te permite guardar verdadero o falso. Ahora también tenemos las estructuras de datos. Esos datos un poco más complejos, formados a partir de los datos primitivos. Por ejemplo, tenemos los objetos, mapas o diccionarios que nos permiten crear una relación de un dato a otro. Por ejemplo, aquí podemos relacionar un texto con otro texto. También un número con un texto. También un texto con un número o un texto con un buleano. Por ejemplo, puedo utilizar el código USA para relacionarlo con el texto Estados Unidos. Ahora, otra estructura de datos son las listas o arreglos. Y también están compuestas de otros datos primitivos. Por ejemplo, puedes tener una lista de textos, puedes tener una lista de números, puedes tener una lista de buleanos, puedes tener una lista de elementos mixtos o puedes tener una lista de listas. Las opciones son ilimitadas. Pero como ves, aquí hemos logrado guardar estos datos en nuestro computador y así vamos a poder trabajar con estos datos. Aquí también debes conocer las constantes. Es son como las variables con la única diferencia de que tú no puedes cambiar estos datos en un futuro. Por ejemplo, el valor de P es un valor universal que no puedes cambiar. Ahora veamos otro concepto que son las funciones. Y piensen las funciones simplemente como un paquete de instrucciones, en donde puedes incluir todo el código que quieras. También aceptan ingresar datos y retornar datos. Por ejemplo, aquí tenemos una función que se llama Sumar, que nos permite ingresar dos variables y dentro de esta función sumamos estas variables y la función retorna un resultado. También puedes crear funciones para multiplicar, funciones para remover el usuario de una lista y puedes crear funciones brutalmente complejas. Como por ejemplo, esta que ves aquí es el algoritmo de Dyxtra, que es una función para encontrar el camino más corto entre dos puntos. Ya que dije la palabra algoritmo, debes saber que un algoritmo es simplemente una función. Es decir, es simplemente un grupo de instrucciones. Ahora veamos los condicionales. Los condicionales te permiten controlar el flujo de tus instrucciones. ¿Qué quieres decir con flujo de instrucciones? Las instrucciones dentro de las funciones se van leyendo una por una y hay momentos en que pueden haber condiciones distintas. Por ejemplo, puedes decir, si es que este usuario tiene su clave correcta, lo dejo entrar. Pero si no tiene la clave correcta, no lo dejo entrar a Facebook. Entonces tú puedes ver aquí, estás controlando el flujo de tu código, pasado en diferentes condiciones. Ahora veamos los ciclos o bucles. Esta herramienta de programación te permite repetir acciones con poco código. Imagina que quiero mostrar un millón de mensajes en WhatsApp, escribir el mismo código un millón de veces no es una buena idea. Entonces los ciclos nos permiten poner simplemente imprime este mensaje en la pantalla para este otro mensaje. Y simplemente le pasas una lista de un millón de mensajes y va a realizar la misma acción para cada uno de ellos. Un ejemplo simple es añadir el número 5 a una lista de números. Y así añade 5 a cada uno de estos números sin hacerlo manualmente uno por uno. Y para empezar te recomiendo que solo aprendas un lenguaje de programación, y así no te vas a confundir con las syntaxes. Las syntaxes es simplemente la forma en que se escribe un lenguaje de programación. De la misma manera que el inglés tiene ciertas palabras, y el español tiene otras palabras, nada lenguaje de programación, a tener una gramática distinta para escribir exactamente los mismos conceptos que hemos visto. Por ejemplo aquí puedes ver el código para imprimir un hola mundo en JavaScript. Ahora este es el mismo código en Java, y este es el mismo código en Python y en C-sharp. Y te preguntas qué es un hola mundo. Es simplemente lo que los programadores llamamos a la primera línea de código que tú escribes al aprender un lenguaje. También debes saber que hay lenguajes tipados y notipados. Los tipados requieren que indiqués el tipo de dato todo el tiempo. Por ejemplo si es que una variable es un texto, es un número, es un buleno, y te ayudan a prevenir errores. Los lenguajes notipados no requieren que digas el tipo de dato todo el tiempo. Estos lenguajes lo identifican, pero pueden haber errores. Por ejemplo aquí tienes JavaScript que no es tipado, y tienes TypeScript que es tipado. Los que no son tipados son excelentes para proyectos pequeños, como por ejemplo Páginas Web, pero los tipados son necesarios para proyectos más grandes y serios. Como por ejemplo una aplicación bancaria. Ahora también hay lenguajes interpretados y compilados. Los interpretados los puedes ejecutar, me funcionan inmediatamente. Los compilados requieren que los empaquetes para poder correrlos. Como debes imaginar los interpretados son flexibles y para correr código rápidamente. Como por ejemplo Python y los compilados son mejores para la eficiencia y la seguridad. Entre estos tienes Java y C-sharp. Ahora todo este código lo vas a escribir en Archivos. Y los archivos te permiten ser más organizado con tu código. Por ejemplo, un archivo te permite exportar ese código e importarlo en otro archivo. Así no tienes un archivo con un millón de líneas. Y con muchos de estos archivos se crean las bibliotecas. Y en línea vas a encontrar millones de bibliotecas que son creadas por diferentes desarrolladores. Tú las puedes utilizar en tus proyectos. Y con eso ya has aprendido todos los conceptos de la lógica de programación. Ahora que ya sabes cómo programar, simplemente escogé un lenguaje de programación. Y te invito a ver este playlist que tiene varios tutoriales de 10 minutos de diferentes lenguajes. Para que puedas aprender realmente rápido. Después de aprender tu primer lenguaje, solo te queda practicar y crear proyectos. Y ya con estos conocimientos puedes crear lo que quieras. Tú creatividad es el límite. Y si en algún momento estás listo para tomar este aprendizaje en serio y deseas utilizarlo de manera profesional, es decir trabajar como programador. Te invito a visitar mis cursos de programación en academyagu.com. Aquí realizamos proyectos del mundo real. Tienes una comunidad con miles de estudiantes. Aprenderás a resolver decenas de algoritmos. Y te preparo para entrevistas con los mismos conocimientos que me permitieron trabajar en proyectos para Google, Nintendo y como ingeniero en Amazon. Estudias o no estudias conmigo. Espero que este contenido sea de gran ayuda para ti. Y te deseo lo mejor en el campo de la programación. Si te gustó este video, te dejo con este otro que confío en que te va a gustar.

El texto ya está en español.