



Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Curso: Inteligencia Artificial

II Semestre 2023

Proyecto Final: Avance

Estudiante:

Celina Madrigal Murillo – 2020059364

Profesor:

Luis Alexander Calvo Valverde

Grupo: 2

Fecha de entrega:

Martes 10 de octubre del 2023

Generador de Historias Ilustradas

Resumen

El tema por desarrollar es un Generador de Historias Ilustradas que combina el lenguaje natural, la generación de imágenes y la traducción automática. La plataforma permitirá que los usuarios proporcionen una sinopsis breve de una historia con detalles clave y que luego esta se expanda utilizando el modelo ChatGPT. Posteriormente, se generará una imagen alusiva a la historia utilizando Stable Diffusion. Además, se integrará DeepL (se cambió a Google Translate) para traducir las historias a ingles haciéndoles más accesibles. La plataforma se implementará en PythonAnywhere para poder ser ejecutada en la nube.

Avances a la fecha

1. Se investigó como se obtienen las APIs de ChatGPT. Para obtener una hay que registrarse o iniciar sesión en OpenAI platform, luego ir a View API keys y generar una. Al intentar hacer pruebas se notó que las APIs de ChatGPT tienen un modelo de precios que generalmente se basa en el consumo de tokens. Al crear una cuenta nueva esta incluye 5 dólares de crédito gratuito con vigencia de 3 meses. Para el proyecto se ha creado una cuenta desde cero, de ser necesario más crédito del que ya se incluye gratuitamente más adelante se valorará pagar lo requerido o crear una cuenta nueva.
2. Se investigó como utilizar la API de ChatGPT en Python. Para poder hacer uso de ella primero se tuvo que instalar una biblioteca llamada openai. Luego de esto se creó un código sencillo para probar el funcionamiento de la API. Primero se configura la clave del API, luego se crea una función para interactuar con la API y luego se utiliza la función con su respectivo

prompt. El código realizado es el siguiente incluyendo su respectivo resultado:

```
import openai

# Configurar clave de API de OpenAI
openai.api_key = 'sk-03HP0f7Jj8zyn8pLORHHT3B1bkFJ0iHxODbgwDbBqwmux9tG'

# Interactuar con la API de ChatGPT
def obtener_respuesta(prompt1):
    response = openai.Completion.create(
        engine="text-davinci-003", #motor
        prompt=prompt1,
        max_tokens=2048
    )
    return response.choices[0].text

# Usar la función para obtener respuestas
prompt1 = "¿Que es ChatGPT?"
respuesta = obtener_respuesta(prompt1)
print(respuesta)
```

```
PS C:\Users\celin\OneDrive\Escritorio\Proyecto IA\Avance> & C:/Users/celin/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/celin/OneDrive/Escritorio/Proyecto IA/Avance/prueba.py"
```

```
ChatGPT es una herramienta de inteligencia artificial creada por OpenAI para entablar conversaciones humanas. Esta herramienta usa un modelo de procesamiento de lenguaje natural (NLP) entrenado para generar respuestas naturales y humanas a través de una conversación simple como un chat.
```

```
Esto significa que ChatGPT puede conversar con los usuarios, aprender nuevos conceptos, y responder preguntas según su memoria y su conocimiento previo. Esto le permite a los desarrolladores mejorar su experiencia de usuario e interactuar más con los usuarios de manera más natural.
```

3. Se investigó como obtener la API de Stable Diffusion. Se ha decidido utilizar Hugging Face, la cual es una comunidad de ciencia de datos, modelos e inteligencia artificial, que ha desarrollado el modelo de Difusores que permite el uso de modelos basados en difusión como la de Stable Diffusion. Se tomó esta decisión ya que es gratuito, simple y más accesible para lograr el objetivo del proyecto. Para ejecutar el modelo de Stable Diffusion solamente es necesaria la instalación de algunas bibliotecas.
4. Se investigó como utilizar el modelo de Stable Diffusion de Hugging Face. Al igual que con la API de ChatGPT se creó un código para comenzar a

hacer pruebas. Primero se importan bibliotecas, luego se define el identificador del modelo pre-entrenado que se utilizará, se ajusta la resolución de la imagen, se define el prompt para generar la imagen, se genera la imagen y por último se muestra. Para ejecutar el código y mostrar la imagen se utilizó un cuaderno de Google Colab. A continuación, se muestra el código creado con su respectivo resultado:

```
import torch

from diffusers import StableDiffusionPipeline, DPMSolverMultistepScheduler

model_id = "stabilityai/stable-diffusion-2-1"

# Use the DPMSolverMultistepScheduler (DPM-Solver++) scheduler here instead
pipe = StableDiffusionPipeline.from_pretrained(model_id, torch_dtype=torch.float16)
pipe.scheduler = DPMSolverMultistepScheduler.from_config(pipe.scheduler.config)
pipe = pipe.to("cuda")

prompt = "un astronauta montando a caballo"

image = pipe(prompt).images[0]

image
```



Se utilizó el siguiente video como guía: [Cómo crear imágenes con IA desde Python | TUTORIAL de Stable Diffusion - YouTube](#)

5. Se investigó como se obtiene la API de DeepL pero como es de pago se decidió cambiar a la de Google Translate la cual utiliza un modelo de aprendizaje automático para traducir entre más de 100 idiomas. Para utilizar esta API se debe instalar la biblioteca googletrans. A continuación, se muestra el código realizado para hacer pruebas con su respectivo resultado:

```
from googletrans import Translator

translator = Translator()

historia = "hola"

resultado = translator.translate(historia, src="es", dest="en")
print(resultado.text)
```

```
PS C:\Users\celin\OneDrive\Escritorio\Proyecto IA\Avance> &
in/OneDrive/Escritorio/Proyecto IA/Avance/prueba.py"
hello
PS C:\Users\celin\OneDrive\Escritorio\Proyecto IA\Avance>
```

Se utilizó el siguiente video como guía: [Google Translate API with Python - YouTube](#)

6. Se unió el código creado para probar la API de ChatGPT y el creado para probar la de Google Translate. Se modificaron los códigos para poder unirlos y a partir de una sinopsis dada por el usuario se genera la historia y esta se traduce a inglés. A continuación, se muestra el código con su respectivo resultado:

```

import openai
from googletrans import Translator

# Configurar clave de API de OpenAI
openai.api_key = 'sk-03HPOf7Jj8zyn8pLORHHT3B1bkFJ0iHxODbgwDbBqwmux9tG'

# Interactuar con la API de ChatGPT
def obtener_respuesta(prompt1):
    response = openai.Completion.create(
        engine="text-davinci-003", #motor
        prompt=prompt1,
        max_tokens=2048
    )
    return response.choices[0].text

# Usar la función para obtener respuestas
print()
historia = input("Dame la sinopsis de tu historia: ")
print()
prompt1 = "Crea una historia de " + historia
respuestaEs = obtener_respuesta(prompt1)
print("Historia en español")
print(respuestaEs+"\n")

translator = Translator()

respuestaEn = translator.translate(respuestaEs, src="es", dest="en")
print("Historia en inglés\n")
print(respuestaEn.text+"\n")

```

Dame la sinopsis de tu historia: un niño de 7 años de Cartago Costa Rica que de mayor quiere ser bombero

Historia en español

Noel, un joven de 7 años de Cartago Costa Rica, siempre había soñado con convertirse en bombero cuando fuera grande. Desde muy pequeño, siempre disfrutó de su imaginación, y a menudo se pasaba el día haciendo dibujos de hermosos carros de bomberos.

Su pasión por volverse bombero fue creciendo durante los años. A medida que se acercaba el cumpleaños número 8, su determinación de alcanzar su sueño fue fortaleciéndose. Y así incitó a su madre a alentarle para seguir su pasión.

Su madre, entusiasmada por ver su determinación, se acercó a algunos profesores de bombero locales y les preguntó si su hijo podría unirse a un grupo de jóvenes de la comunidad que estaban preparando para el oficio. Los profesores vieron su entusiasmo y le dijeron que, si trabajaba duro, podría alcanzar su objetivo.

Y eso es exactamente lo que hizo. Con una gran cantidad de dedicación para trabajar duro, Noel fue acercándose a su sueño de convertirse en bombero. Dedicó meses al estudio de la materia y a la práctica de diversas habilidades y competencias especiales.

A la edad de 18 años, Noel estaba listo para unirse al servicio de bomberos. Estaba emocionado y ansioso para comenzar una vida de trabajo y servicio como bombero.

Todos sus esfuerzos ahora se han visto recompensados. Noel ha servido en el departamento de bomberos durante los últimos cinco años, y es uno de los bomberos más respetados y queridos de Cartago. Él es un ícono local, y un ejemplo a ser seguidor para todos los demás jóvenes de la ciudad que tienen sueños de convertirse en bomberos, igual que él hizo.

Historia en inglés

Noel, a 7-year-old boy from Carthage Costa Rica, had always dreamed of becoming a firefighter when he was big. From an early age, he always enjoyed his imagination, and often spent the day making drawings of beautiful firefighters.

His passion for becoming a firefighter was growing over the years. As the 8th birthday approached, his determination to achieve his dream was strengthening himself. And thus prompted his mother to encourage him to follow his passion.

His mother, excited to see his determination, approached some local firefighter teachers and asked them if their son could join a group of young people from the community who were preparing for the trade. The teachers saw their enthusiasm and told him that, if he worked hard, he could achieve his goal.

And that is exactly what he did. With a lot of dedication to work hard, Noel approached his dream of becoming a firefighter. He dedicated months to the study of the subject and the practice of various special skills and competences.

At the age of 18, Noel was ready to join the Firefighters service. I was excited and anxious to start a work and service life as a firefighter.

All his efforts have now been rewarded. Noel has served in the fire department for the past five years, and is one of the most respected and beloved firefighters in Carthage.

He is a local icon, and an example to be a follower for all other young people in the city who have dreams of becoming firefighters, just as he did.

7. Se creó una cuenta gratuita en PythonAnywhere llamada Beginner account la cual tiene ciertas limitaciones. De ser necesario se podría pagar la cuenta Hacker la cual tiene un costo de 5 dólares al mes.
8. Se creó la Web App en PythonAnywhere. Se creó con Flask ya que se considera lo mejor para el propósito del proyecto. El enlace es el siguiente:
<http://celina19.pythonanywhere.com/>

Cronograma del trabajo faltante

Las tareas pendientes son las siguientes:

- Estudiar a más profundidad cómo funciona y como se utiliza PythonAnywhere.
- Integrar las 3 funcionalidades (generación de la historia, generación de la imagen y la traducción de la historia) para que trabajen de manera conjunta.
- Crear un interfaz sencilla y amigable para la aplicación.
- Realizar el video de la aplicación y el artículo solicitado.

A continuación, se presenta el cronograma de las siguientes semanas para esas actividades:

Actividad	Fechas
Estudio PythonAnywhere	Del 11/10/2023 al 18/10/2023

Integrar las 3 funcionalidades	Del 19/10/2023 al 26/10/2023
Crear la interfaz	Del 27/10/2023 al 03/11/2023
Realizar el video y el artículo	Del 04/11/2023 al 6/11/2023