

Generador de resumen de páginas web

Andrés Alejandro Jiménez López

Contexto

A continuación, se mostrará el proceso necesario para generar un código funcional en lenguaje Python, que genere un resumen de una página web teniendo únicamente como dato de entrada la url de la misma, todo esto mediante el uso de prompt engineering y ChatGPT.

Primera pregunta

A

Dame el código en python necesario para hacer el resumen de un texto hasta el punto de cargar la pipeline, haciendo uso de la biblioteca Transformers de hugging face y específicamente el modelo de facebook /bart-large-cnn

Respuesta

```
!pip install transformers

from transformers import pipeline

# Cargar el modelo de resumen de BART
resumen = pipeline("summarization", model="facebook/bart-large-cnn")
```

Segunda pregunta

A

¿Cómo se obtiene y almacena el contenido de un blog desde la URL proporcionada en lenguaje Python. Haciendo uso de las etiquetas de encabezado (h1) y parrafo (p) que se pueden encontrar por el lenguaje html con el que está creado el blog?

Respuesta

```
!pip install requests beautifulsoup4
import requests
from bs4 import BeautifulSoup
# Definir la URL del blog
URL = "https://www.xataka.com/analisis/iphone-15-pro-max-opiniones-toma-contacto-video-fotos"

# Realizar una solicitud HTTP para obtener el contenido de la página
response = requests.get(URL)

# Verificar si la solicitud fue exitosa (código de respuesta 200)
if response.status_code == 200:
    # Parsear el contenido HTML de la página con BeautifulSoup
    soup = BeautifulSoup(response.text, 'html.parser')

    # Extraer encabezados (h1) y párrafos (p) del blog
    encabezados = soup.find_all('h1')
    parrafos = soup.find_all('p')

    # Crear una variable para almacenar el contenido del blog
    contenido_blog = ""

    # Agregar encabezados al contenido del blog
    for h1 in encabezados:
        contenido_blog += h1.get_text() + "\n"

    # Agregar párrafos al contenido del blog
    for p in parrafos:
        contenido_blog += p.get_text() + "\n"

    # Ahora, contenido_blog contiene el contenido del blog con encabezados (h1) y párrafos (p)
else:
    # En caso de que la solicitud no sea exitosa, manejar el error
    print("Error al obtener la página:", response.status_code)
```

Tercera pregunta

A

¿Puedes proporcionarme un código en Python que dado un artículo que está almacenado en la variable "contenido_blog", divida el artículo en fragmentos de hasta 300 caracteres?

Respuesta

```
fragmentos = []
fragmento_actual = ""
longitud_maxima = 300

for palabra in contenido_blog.split():
    if len(fragmento_actual) + len(palabra) + 1 <= longitud_maxima: # +1 para el espacio
        if fragmento_actual:
            fragmento_actual += " "
        fragmento_actual += palabra
    else:
        fragmentos.append(fragmento_actual)
        fragmento_actual = palabra

if fragmento_actual:
    fragmentos.append(fragmento_actual)

# Ahora, 'fragmentos' contendrá los fragmentos de hasta 300 caracteres
for i, fragmento in enumerate(fragmentos, start=1):
    print(f"Fragmento {i}: {fragmento}")
```


Cuarta pregunta

A

proporcióname el código en Python que muestre cómo extraer resúmenes de una lista de diccionarios con un campo 'summary_text' y luego compilar todos los resúmenes en una sola cadena"

Respuesta

```
resumenes = resumen(fragmentos, max_length=120, min_length=30, do_sample=False)

# Extraer los resúmenes de los diccionarios
Ex_resumenes = [item['summary_text'] for item in resumenes]

# Compila todos los resúmenes con espacios
texto_resumen = " ".join(Ex_resumenes)
```

Quinta pregunta

A

proporcióname el código en python que me permita exportar un texto almacenado en la variable "texto_resumen" a un archivo .txt

Respuesta

```
with open('resumenblog2.txt', 'w', encoding='utf-8') as archivo:  
    archivo.write(texto_resumen)
```