Overview

This project consists of two main parts:

1. **Web Scraping and Data Extraction**
2. **Transformer-Based Chatbot Development and Evaluation**

The aim is to extract product data from an e-commerce website, store it for analysis, and enhance user interaction with a chatbot built using transformer models.

Part 1: Web Scraping and Data Extraction

Key Features:

- **Libraries Used**:
    - requests, BeautifulSoup, pandas, json, datetime, os, time, re.
- **Scraper Class**:
    - **DrRashelScraper**: A Python class designed to scrape product information from the Dr. Rashel website.
    - Extracts product details such as name, price, SKU, description, categories, images, and specifications.
    - Collects site metadata, contact details, and social media links.
- **Data Saving**:
    - Stores scraped data in Google Drive in JSON and CSV formats.
    - Generates a summary of the scraping session.

How to Use:

1. Install the required libraries:

bash

Copy code

pip install requests beautifulsoup4 pandas

2. Mount Google Drive in Google Colab:

python

Copy code

from google.colab import drive

drive.mount('/content/drive')

3. Execute the script to scrape the data:

python

Copy code

```
url = "https://dr.rashel.in/"
scraper = DrRashelScraper(url)
data = scraper.scrape_website()
if data:
  save_to_drive(data, 'dr_rashel_data')
```

4. Scraping results include:
    - Complete data in JSON format.
    - Product data in CSV format.
    - Summary of scraping in JSON.

Part 2: Transformer-Based Chatbot Development

Key Features:

- **Libraries Used**:
    - transformers, pandas, datetime.
- **LLM-Based Chatbot**:
    - Uses GPT-2 for text generation to simulate chatbot responses.
- **Event Tracking**:
    - Tracks user events such as product views, messages, and purchases.
    - Logs user behavior for analysis.

Workflow:

1. Install the required library:

bash

Copy code

```
pip install pandas transformers
```

2. Create product data and event logs for analysis.
3. Analyze user behavior:

python

Copy code

```
analyze_user_behavior(df_events)
```

4. Generate chatbot responses:

python

Copy code

```
chatbot_reply = generate_chatbot_response("What products help with acne?")
print(f"Chatbot Response: {chatbot_reply}")
```

Project Structure

- **DrRashelScraper**: Handles scraping tasks.
- **save_to_drive**: Saves extracted data in JSON/CSV format.
- **analyze_user_behavior**: Tracks user activity and provides insights.
- **generate_chatbot_response**: Simulates chatbot responses for user queries.

Dependencies

Ensure the following Python libraries are installed:

- requests
- BeautifulSoup4
- pandas
- transformers
- datetime
- json

Use Cases

1. **E-commerce Insights**:
    - Extract and analyze product data.
    - Understand user behavior through event tracking.
2. **Chatbot Enhancement**:
    - Utilize GPT-2 to respond to user queries effectively.

Future Work

- Integrate more advanced transformer models like GPT-3 or custom fine-tuned models.
- Improve scraping robustness for dynamic websites using Selenium or Playwright.
- Add visualization tools for user behavior analysis.