Web Scraping and GPT Model Training

Code Files

- **Hospital.py**: This file includes the function hospital_list() which is responsible for collecting hospital URLs.
- **Scrap.py**: This script handles the scraping of hospital websites, saving the gathered data into a file named scraped_json.json.
- **model.ipynb**: This notebook encompasses the entire data preprocessing, model training, and saving process for the GPT model.

Data Files

- **scraped_data.json**: This file stores the data collected from the websites of top hospitals.

Trained Model

The finalized Private GPT model has been stored in the directory./gpt_finetuned.

Data Collection

In the initial code file, **Hospital.py**, the function hospital_list() is responsible for extracting the top hospital URLs from a table on the Newsweek website. The links are gathered and saved into a list, which is returned as output.

To gather data from the websites of the top 50 hospitals, a web scraping process was set up using the requests library to fetch HTML content and BeautifulSoup for parsing it.

- **Note**: Only data inside the <p> tags were scraped to focus on key information and improve efficiency.
- **Note**: For faster scraping, the depth of the links followed by the scraper was set to 0, though this can be adjusted as required.

Data Preprocessing

Data preprocessing took place within the **model.ipynb** file. The main objective was to clean the scraped data by eliminating unwanted characters, whitespace, and escape sequences. Regular expressions were applied to perform these cleanup operations. Afterward, the data was formatted appropriately for the training phase of the model.

Model Training

In the **model.ipynb** file, the cleaned data was used to train a Private GPT model. The transformers library, which provides tools for working with state-of-the-art machine learning models, was used throughout the process.

Key Steps Involved in Model Training:

- **Data Preparation**:
    - The text data was tokenized using the GPT2 tokenizer from the transformer's library.
- **Model Configuration and Training**:
    - The model chosen for training was the pre-trained GPT2 from Hugging Face.
    - The model was transferred to a GPU for faster computation using .to('cuda').
    - Training progress was monitored and logged using **Wandb**, a platform designed to track and visualize machine learning experiments. The WandbCallback() was integrated to log training metrics in real-time.
    - After training completion, the final trained model checkpoint was saved.

Conclusion

This project successfully met the goal of scraping relevant data from top hospital websites and utilizing this data to train a Private GPT model. The entire process, from data collection to model training, was completed effectively, with the model being stored for further use and analysis.