NLP Tasks Project Overview

Memory updated

Here are the **stages for the text summarization project** using BART, tailored for your GitHub repository setup:

Project Folder Structure:

bash

```
NLP-Text-Summarization/
│
├── Data/
│   └── Medical_book.pdf     # The medical textbook
│
├── app/
│   ├── __init__.py          # For app initialization
│   ├── app.py               # Flask/Streamlit app backend
│   └── templates/
│       └── index.html       # Frontend UI (if using Flask)
│
├── models/
│   └── fine_tuned_bart/     # To store fine-tuned BART model after training
│
├── notebooks/
│   └── data_preprocessing.ipynb   # Jupyter notebook for text extraction and
preprocessing
│   └── fine_tuning_bart.ipynb     # Jupyter notebook for model fine-tuning and evaluation
│
├── requirements.txt         # Required Python libraries
├── README.md                # Project description
└── .gitignore               # Files to ignore (e.g., models)
```

Stage 1: Dataset Preparation

Steps:

1. **Text Extraction from Medical Book**:
   - Write a script/notebook to extract text from `Medical_book.pdf` located in the **Data/** folder.
   - Clean and preprocess the extracted text (remove unwanted characters, line breaks, etc.).
   - **File**: `notebooks/data_preprocessing.ipynb`

2. **Segmentation and Pairing**:
   - Divide the extracted text into meaningful sections (e.g., disease descriptions, diagnosis sections).
   - Pair the text chunks with their corresponding summaries (either manually or using summary-like sections from the book).

   Example of pairing:

```python
{'text': 'Diabetes is a chronic disease...', 'summary': 'Diabetes causes high blood sugar...'}
```

Stage 2: Model Selection and Fine-Tuning

Steps:

1. **Set Up the Environment**:
    - Create a **Python virtual environment**:

      ```bash
      python -m venv nlp_project
      source nlp_project/bin/activate  # Linux/Mac
      nlp_project\Scripts\activate  # Windows
      ```

    - Install required packages listed in `requirements.txt` (add packages like **transformers**, **datasets**, **torch**):

      ```bash
      pip install -r requirements.txt
      ```

2. **Fine-Tuning BART**:
    - Use **BART (facebook/bart-large-cnn)** for fine-tuning.
    - Write a **Jupyter notebook** (`notebooks/fine_tuning_bart.ipynb`) to fine-tune BART on your segmented medical text dataset.
    - Save the fine-tuned model in the **models/** folder.

3. **Save Fine-Tuned Model**:
    - After fine-tuning, save the model to the **models/fine_tuned_bart/** directory for future inference.

Stage 3: Model Evaluation

Steps:

1. **Evaluate the Model**:
    - Use evaluation metrics like **ROUGE** and **BLEU** to evaluate the model's performance on the test dataset.
    - Write the evaluation code in the `notebooks/fine_tuning_bart.ipynb` notebook.

2. **Store Evaluation Results**:

  - Save the evaluation metrics and model performance results.

Stage 4: App Development

Steps:

1. **Set Up Flask/Streamlit App**:
   - Use **Flask** or **Streamlit** to create a web app.
   - The app will take input medical text from users and display the summarized output using the fine-tuned BART model.

2. **Build the UI**:
   - Create a basic web interface using HTML (for Flask) or Streamlit's UI components.
   - **File**: `` `app/templates/index.html` `` (Flask) or `` `app/app.py` `` (Streamlit).

3. **Integrate the Model**:
   - Load the fine-tuned BART model from **models/fine_tuned_bart/**.
   - When the user submits medical text, pass it to the model, and return the generated summary.

4. **Run the App**:
   - Use Flask or Streamlit to serve the app:

     ```
     bash
     ```

     ```
     python app/app.py
     ```

Stage 5: Testing and Deployment

Steps:

1. **Test the App**:
   - Test the app by inputting various sections of the medical textbook and evaluating the quality of the generated summaries.

2. **Deploy the App**:
   - Deploy the web app on platforms like **Heroku**, **Streamlit Cloud**, or **AWS** for public access.

Stage 6: Version Control with GitHub

Steps:

1. **Add Files to Git**:
   - Track your project files using Git:

bash

```
git add .
git commit -m "Initial commit with dataset and model"
```

2. **Push to GitHub**:
   - Push your local changes to the remote repository:

     bash

     ```
     git push origin main
     ```

Summary of the Plan:

- **Stage 1**: Prepare the dataset (extract and preprocess text from the medical book).
- **Stage 2**: Fine-tune the **BART** model for text summarization.
- **Stage 3**: Evaluate the model performance using ROUGE/ BLEU.
- **Stage 4**: Build a web app (Flask/Streamlit) for summarizing input text.
- **Stage 5**: Test and deploy the app.
- **Stage 6**: Use GitHub for version control and storing the project.

Let me know if this plan is good to proceed!