Documentation

Blog Generation with Llama 2 and Streamlit

Overview:

The Blog Generation application. This project leverages the Llama 2 language model, based on the transformer architecture, to generate blog content. The user interface is built using Streamlit, making it easy to interact with the model and generate blog responses.

Topics:

Installation:

To run the Blog Generation application, follow these steps:

1. Clone the repository:

   bash
   git clone https://github.com/BhanukiranG/Blog-Generation

2. While running the application
Download model [TheBloke/Llama-2-7B- Chat-GGML] from hugging face named [llama-2-7b-chat.ggmlv3.q8_0.bin]
https://huggingface.co/TheBloke/Llama-2-7B-Chat-GGML/tree/main
Download [python==version]

3. Install dependencies:

   pip install -r requirements.txt

Running the Application:

Execute the following command to start the Streamlit app:
streamlit run app.py

User Interface:

1. Enter the Blog Topic:
   - Input the main topic of the blog.

2. Additional Fields:
   - Specify the number of words and select the target audience.

3. Generate Button:
   - Click the "Generate" button to initiate blog generation.

4. View Generated Blog:
   - The generated blog will be displayed on the app.

Customization:

Model Configuration:

Adjust the Llama 2 model configuration in the CTransformers initialization in the getLLamaresponse function. Tune parameters like max_new_tokens and temperature for different results.

Python:

```
llm = CTransformers(model='models/llama-2-7b-chat.ggmlv3.q8_0.bin',
                    model_type='llama',
                    config={'max_new_tokens': 256,
                            'temperature': 0.01})
```

Template Modification:

Modify the prompt template in the getLLamaresponse function for customized prompts:

Python:

```
template = """
    Write a blog for {blog_style} job profile for a topic {input_text}
    within {no_words} words.
        """
```

Presentation:

- **Title:** Blog Generation with Llama 2 and Streamlit
- **Subtitle:** Leveraging Large Language Models for Seamless Content Generation

Project Overview

- **Objective:**
  - Generate blog content using Large language models.
- **Components:**

1. **Llama 2 Language Model:**
   - A transformer-based language model.
   - Capable of understanding and generating human-like text.
2. **Streamlit Web Application:**
   - A user-friendly framework for building interactive web applications.
   - Enables seamless interaction with the Llama 2 model.

Llama 2 Language Model

- **Key Features:**
  1. **Transformer Architecture:**
     - Allows capturing long-range dependencies in sequences.
  2. **Self-Attention Mechanism:**
     - Weights different parts of the input sequence differently during processing.
  3. **Multi-Head Attention:**
     - Captures different aspects of the input sequence through multiple attention heads.

Streamlit Web Application

- **Purpose:**
  - Provide an intuitive interface for users to interact with the language model.
- **Streamlit Features:**

1. **Easy-to-Use Interface:**
   - Simplifies the process of interacting with the model.
2. **User Input Fields:**
   - Collects input for generating blog content.
3. **Generate Button:**
   - Initiates the blog generation process.
4. **Dynamic Output:**
   - Displays the generated blog content in real-time.

Installation

Running the Application

- **Command:**
  - Execute **streamlit run app.py** in the terminal.
- **Access:**
  - Open the provided link in the terminal to access the application.

User Interface

- **Input Fields:**
  1. **Blog Topic:**
     - Main topic for the generated blog.
  2. **Number of Words:**
     - Specifies the desired word count.
  3. **Target Audience/Job Profile:**
     - Selects the audience for the generated blog.
- **Generate Button:**
  - Initiates the blog generation process.

Customization

- **Model Configuration:**
  - Tune parameters such as **max_new_tokens** and **temperature** in the **CTransformers** initialization.
- **Template Modification:**
  - Customize the prompt template in the **getLLamaresponse** function for tailored blog generation.
  -