LLama3 ReadMe

Anmol Valecha

June 2024

Introduction

This document provides a step-by-step guide for downloading, installing, and running the Llama3 model on an M1 Mac. Additionally, it includes instructions for setting up a Python environment, installing Streamlit, and creating a chatbot to interact with the model.

Prerequisites

- M1 Mac
- Python installed
- Visual Studio Code (VS Code) installed
- Internet connection

Step-by-Step Instructions

1. Setting Up Python Virtual Environment

1. **Create a New Folder**:
   - Create a new folder named `Assignment3` and navigate into it:

     ```
     mkdir Assignment3
     cd Assignment3
     ```

2. **Create a Virtual Environment**:
   - Run the following command to create a virtual environment:

     ```
     python3 -m venv .venv
     ```

3. **Activate the Virtual Environment**:

- Activate the virtual environment with:

  ```
  source .venv/bin/activate
  ```

4. **Install Streamlit**:
   - Inside the activated virtual environment, install Streamlit using pip:

     ```
     pip install streamlit
     ```

2. Download and Install Llama3

1. **Download Ollama from GitHub**:
   - Visit the Llama3 GitHub repository.
   - Download the ZIP file and extract it.
2. **Install Ollama**:
   - Open the terminal and navigate to the extracted folder.
   - Run the following command to install Ollama:

     ```
     ./install.sh
     ```

   - To verify the installation, run:

     ```
     ollama --version
     ```

3. **Run Llama3 Model**:
   - To run the model, use the command:

     ```
     ollama run phi3
     ```

   - This will download and start the model in the terminal.
4. **Check if Llama3 is Running**:
   - Open a new terminal window and run:

     ```
     lsof -i tcp:11434
     ```

   - This command checks if the Llama3 service is running on port 11434.
5. **Close the Chat**:
   - To close the chat, use the following command in the terminal:

     ```
     /bye
     ```

3. Code for the Chatbot

Listing 1: chatbot.py

```
import streamlit as st
import requests
import json

# Function to convert session history to messages format
def convert_to_messages(history):
    messages = []
    for entry in history:
        if entry.startswith("You:"):
            messages.append({"role": "user", "content": entry[5:]})
        elif entry.startswith("Bot:"):
            messages.append({"role": "assistant", "content": entry[5:]})
    return messages

# Function to send request to the local model service
def get_model_response(prompt):
    url = "http://localhost:11434/api/chat"
    headers = {"Content-Type": "application/json"}
    messages = convert_to_messages(st.session_state.conversation)

    data = {
        "model": "phi3",
        "messages": messages,
        "stream": False
    }

    response = requests.post(url, headers=headers, data=json.dumps(data))

    try:
        response_json = response.json()
        return response_json['message']['content']
    except json.JSONDecodeError as e:
        st.error(f"JSON decode error: {e}")
        return "Error: Invalid response from the model service."

# Streamlit app layout
st.title("Chat with Local LLM")

# Initialize conversation history
if 'conversation' not in st.session_state:
    st.session_state.conversation = []
```

```
# User input
user_input = st.text_input("You:", key="user_input")

# Check if user input is provided and the Send button is clicked
if st.button("Send"):
    if user_input:
        # Add user input to conversation history
        st.session_state.conversation.append(f"You: {user_input}")

        # Get model response
        response = get_model_response(user_input)

        # Add model response to conversation history
        st.session_state.conversation.append(f"Bot: {response}")

        # Clear the user input box by resetting its state
        user_input = ""

# Display current interaction (You: input / Bot: response)
if st.session_state.conversation:
    latest_interaction = st.session_state.conversation[-2:]
    st.subheader("Current Interaction")
    for message in latest_interaction:
        st.write(message)

# Display conversation history
st.subheader("Conversation History")
conversation_history = "\n".join(st.session_state.conversation)
st.text_area("History", value=conversation_history, height=300, disabled=True)

# Clear conversation button
if st.button("Clear Conversation"):
    st.session_state.conversation = []
```

4. Running the Chatbot

1. **Start Streamlit**:
   - In the terminal, run:

     ```
     streamlit run chatbot.py
     ```

2. **Interact with the Model**:
   - Open the provided local URL in a web browser.
   - Interact with the chatbot interface, which sends requests to the Llama3 model running locally.

5. Setting Up Llama3 User Interface

1. **Download the UI**:
   - Follow the instructions provided in the Llama3 GitHub repository to download the user interface.
2. **Install Dependencies**:
   - Navigate to the UI folder and install the required dependencies:

     ```
     cd llama3-ui
     npm install
     ```

3. **Start the User Interface**:
   - Run the UI:

     ```
     npm start
     ```

4. **Access the UI**:
   - Open the provided local URL in a web browser to access the Llama3 user interface.

Example Chat Interactions

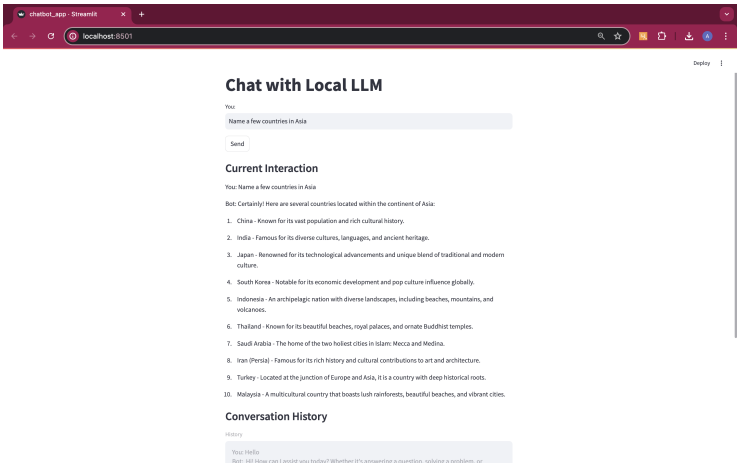

Figure 1: Chat with Ollama phi3 model - Image 1

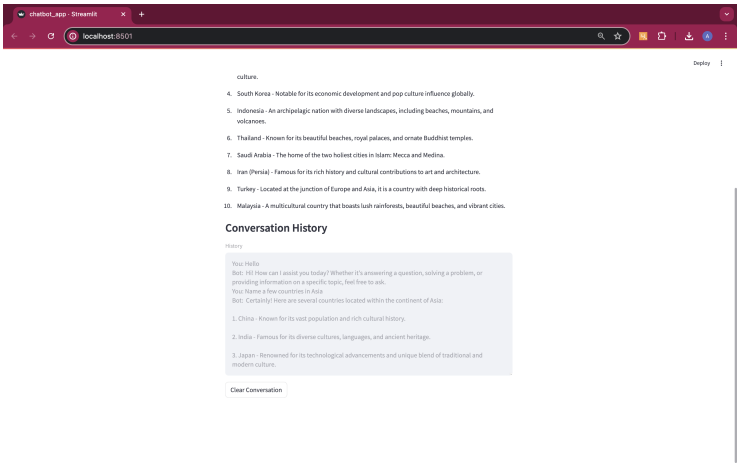

Figure 2: Chat with Ollama phi3 model - Image 2