

Installation and Documentation Manual: Weather-Chatbot-Phi3

Vatsal Patel, MSc Artificial Intelligence (60 ECTS)

Matriculation No: 3190684

23.07.2024

Contents

1	Introduction	3
1.1	Key Features and Innovations	3
1.2	This Manual Provides:	3
1.3	Who Should Use This Manual?	3
2	Installation	3
2.1	Python Package Setup	3
2.1.1	Create a Separate Environment	3
2.1.2	Install Package from PyPI	4
2.1.3	Run Chatbot	4
2.2	Docker Setup	4
2.2.1	Pull Docker Image	4
2.2.2	Run Docker Container	4
2.2.3	Build Docker Image Locally	4
3	Installation from Submitted Files	5
3.1	Build Docker Image from Submitted Files	5
3.2	Install Python Package from Submitted Files	5
4	Resource Requirements	5
5	Usage Instructions	5
5.1	First-Time User Experience Python Pip Package	6
5.1.1	Unix Terminal or Command Prompt	6
5.1.2	Accessing the Web App	6
5.1.3	Testing the Chatbot	7
5.1.4	Shutting Down the Chatbot	7
5.2	User Experience Docker Package	8
5.2.1	After Docker build/pull	8
5.2.2	Accessing the Web App	8
5.2.3	Testing the Chatbot	8
5.2.4	Shutting Down the Chatbot	8
6	Troubleshooting	9

7	Limitations	9
8	Online Extension	9
9	Computational Resources	9

1 Introduction

This manual guides you through the installation and use of a cutting-edge weather forecast chatbot. This chatbot represents a significant advancement in the application of Large Language Models (LLMs) for practical tasks. By harnessing the power of the Phi-3-mini-4k-instruct model and the OpenWeather API, this chatbot delivers real-time, accurate weather forecasts directly through an intuitive web interface.

1.1 Key Features and Innovations

- **Lightweight LLM:** Leverages the Phi-3-mini-4k-instruct model, a smaller LLM that efficiently runs on standard CPUs, demonstrating the expanding capabilities of resource-efficient models.
- **Real-time Weather Data:** Seamlessly integrates with the OpenWeather API to provide up-to-the-minute weather information.
- **User-friendly Interface:** Offers a simple web-based interface for easy interaction and query input.
- **LangChain Integration:** Utilizes LangChain tools and agents, showcasing an innovative approach to building sophisticated LLM applications.

1.2 This Manual Provides:

- **Installation Instructions:** Detailed steps for setting up the chatbot using Docker or Python packages.
- **Usage Guide:** A walkthrough of the chatbot interface and how to get weather forecasts.
- **Troubleshooting Tips:** Solutions for common issues you might encounter.
- **Limitations and Future Directions:** A discussion of current constraints and the exciting potential for future enhancements as CPU-friendly LLMs continue to evolve.

1.3 Who Should Use This Manual?

This manual is designed for anyone interested in exploring the intersection of LLMs and practical applications. Whether you're a developer seeking to understand LLM integration or a user looking for a convenient and accurate weather tool, this chatbot and its accompanying documentation offer valuable insights into the rapidly expanding world of language model technology.

2 Installation

2.1 Python Package Setup

Note: The pip-based installation is preferred over Docker for users with low CPU cores.

2.1.1 Create a Separate Environment

Before proceeding with the installation, it is recommended to create a separate Conda or Python virtual environment:

```
# Using Conda
conda create --name weather-chatbot-env python=3.11
conda activate weather-chatbot-env

# Using Python's venv
python -m venv weather-chatbot-env
source weather-chatbot-env/bin/activate

# On Windows use:
weather-chatbot-env\Scripts\activate
```

2.1.2 Install Package from PyPI

```
pip install weather-chatbot-phi3
```

2.1.3 Run Chatbot

```
weather-chatbot-phi3
```

2.2 Docker Setup

2.2.1 Pull Docker Image

```
docker pull vatsalpatel18/weather-chatbot-phi3
```

2.2.2 Run Docker Container

```
docker run -p 7680:7680 vatsalpatel18/weather-chatbot-phi3
```

2.2.3 Build Docker Image Locally

If you prefer to build the Docker image locally from the provided submission files, follow these steps:

```
# Unzip the submission files
unzip project_code_files_Patel_Vatsal_3190684_Weather-Chatbot.zip

# Navigate to the Docker directory
cd Docker

# Build the Docker image
docker build -t weather-chatbot-local .

# Run the Docker container
docker run -p 7680:7680 weather-chatbot-local
```

3 Installation from Submitted Files

3.1 Build Docker Image from Submitted Files

If you prefer to build the Docker image locally from the provided submission files, follow these steps:

```
# Unzip the submission files
unzip project_code_files_Patel_Vatsal_3190684_Weather-Chatbot.zip

# Navigate to the Docker directory
cd Docker

# Build the Docker image
docker build -t weather-chatbot-local .

# Run the Docker container
docker run -p 7680:7680 weather-chatbot-local
```

3.2 Install Python Package from Submitted Files

If you prefer to install the package from the submitted files, follow these steps:

```
# Unzip the submission files
unzip project_code_files_Patel_Vatsal_3190684_Weather-Chatbot.zip

# Navigate to the package directory
cd Weather-Chatbot-phi3_package

# Install the package using the wheel file
pip install dist/weather_chatbot_phi3-0.1.3-py3-none-any.whl

# Run the chatbot
weather-chatbot-phi3
```

4 Resource Requirements

The following table lists the minimum and preferred resource requirements for running the chatbot:

Requirement	CPU Cores	RAM
Minimum	4	12 GB
Preferred	8	32 GB
Development	16	32 GB

5 Usage Instructions

Once the chatbot is set up, you can interact with it through the specified port (7680). The chatbot can process user queries and provide weather forecasts in real time.

5.1 First-Time User Experience Python Pip Package

5.1.1 Unix Terminal or Command Prompt

After running the following command in your terminal:

```
weather-chatbot-phi3
```

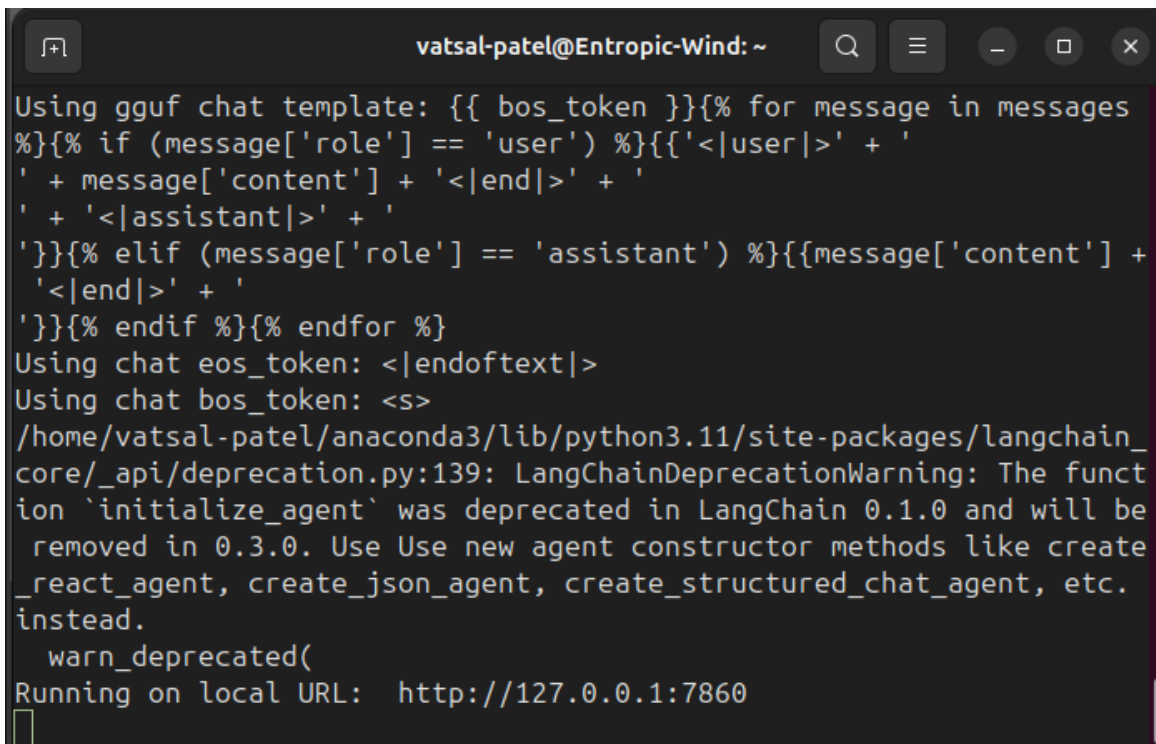
First-time users will see the model being downloaded, as shown below (Figure 1):

```
(base) vatsal-patel@Entropic-Wind:~$ weather-chatbot-phi3
Phi-3-mini-4k-instruct-q4.gguf: 6%|          | 136M/2.39G [00:09<02:14, 16.7MB/s]
```

Figure 1: Model download process for first-time Python package users

Once the model is downloaded, you will see a message indicating the local URL for accessing the chatbot:

```
Running on local URL: http://127.0.0.1:7860
```



The screenshot shows a terminal window titled "vatsal-patel@Entropic-Wind: ~". The output of the command is as follows:

```
Using gguf chat template: {{ bos_token }}{% for message in messages
%}{% if (message['role'] == 'user') %}{{ '<|user|>' + '
' + message['content'] + '<|end|>' + '
' + '<|assistant|>' + '
'}}{% elif (message['role'] == 'assistant') %}{{ message['content'] +
'<|end|>' + '
'}}{% endif %}{% endfor %}
Using chat eos_token: <|endoftext|>
Using chat bos_token: <s>
/home/vatsal-patel/anaconda3/lib/python3.11/site-packages/langchain_
core/_api/deprecation.py:139: LangChainDeprecationWarning: The funct
ion 'initialize_agent' was deprecated in LangChain 0.1.0 and will be
removed in 0.3.0. Use Use new agent constructor methods like create
_react_agent, create_json_agent, create_structured_chat_agent, etc.
instead.
  warn_deprecated(
Running on local URL: http://127.0.0.1:7860
```

Figure 2: Local URL for accessing the chatbot

Click on this URL or paste it into your web browser (Google Chrome, Firefox, or Edge).

5.1.2 Accessing the Web App

After navigating to the local URL, you will see the following web interface (Figure 3):

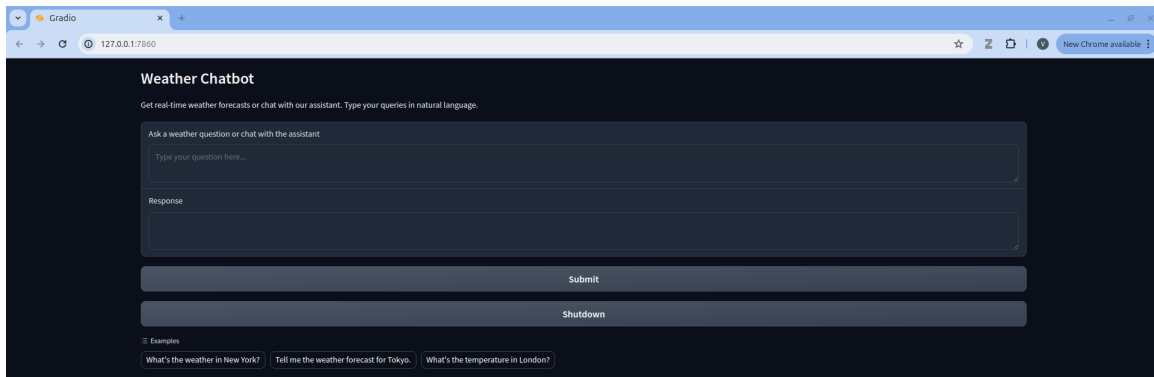


Figure 3: Weather Chatbot web interface

5.1.3 Testing the Chatbot

To test the chatbot, enter a query such as "What's the weather in New York?" in the text box and click on the "Submit" button. The process might take some time depending on your machine. On a developer machine with 16 cores (Ryzen 7) and 16 GB RAM, it takes around 100 seconds.

While processing, you can check the terminal for the backend running LangChain, as shown below (Figure 4):

```
vatsal-patel@Entropic-Wind: ~
> Entering new AgentExecutor chain...

llama_print_timings:      load time =      278.10 ms
llama_print_timings:      sample time =       4.67 ms /   33 runs (
  0.14 ms per token, 7069.41 tokens per second)
llama_print_timings: prompt eval time =    7335.05 ms /  244 tokens (
 30.06 ms per token,   33.26 tokens per second)
llama_print_timings:       eval time =    2184.55 ms /   32 runs (
 68.27 ms per token,   14.65 tokens per second)
llama_print_timings:      total time =    9551.34 ms /  276 tokens

I need to find out today's weather in New York.
Action: WeatherLookup
Action Input: "New York"
Observation: Weather: light rain, Temperature: current 21.8°C, feels like
 22.46°C, min 20.47°C, max 22.84°C, Pressure: sea level 1015 hPa, ground
 level 1013 hPa, Humidity: 93%, Visibility: 10000 meters, Wind: speed 4.92
 m/s, deg 19, Clouds: 100%, Rain: 0.13 mm, Date/Time: 2024-07-23 08:25:22
, Timezone: -14400 seconds, City Name: New York, Response Code: 200
Thought: Llama.generate: prefix-match hit
```

Figure 4: Backend running LangChain

After some time, the response will be generated and displayed on the user interface, as shown below (Figure 5):

5.1.4 Shutting Down the Chatbot

To shut down the running chatbot, simply press the "Shutdown" button on the web interface. This will close the running bot at the backend.

Make sure to close the web browser window after shutting down the chatbot to free up system resources.

This concludes the first-time user experience for the Python Pip Package.

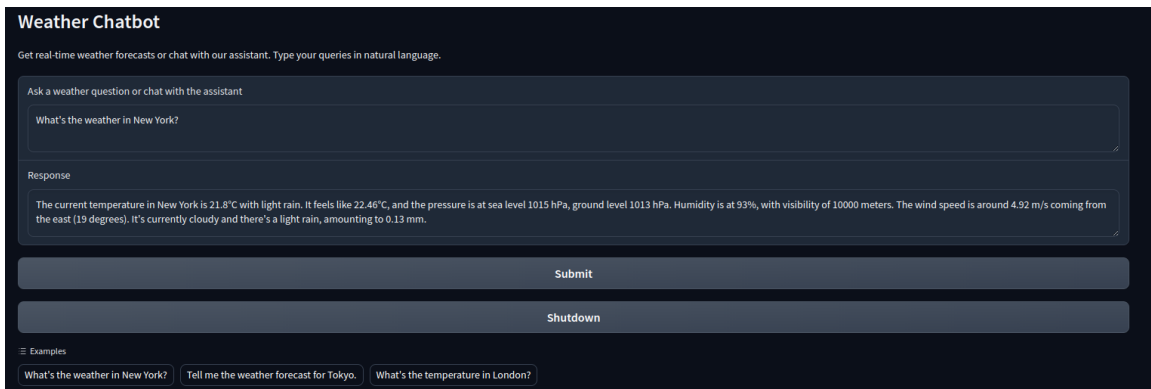


Figure 5: Generated response on the user interface

5.2 User Experience Docker Package

5.2.1 After Docker build/pull

After running the following command in your terminal:

```
docker run -p 7680:7680 vatsalpatel18/weather-chatbot-phi3
```

The Docker container will start the chatbot service, and the model will already be available since it was included in the Docker image.

Once the container is running, you will see a message indicating the local URL for accessing the chatbot:

```
Running on local URL: http://127.0.0.1:7860
```

Click on this URL or paste it into your web browser (Google Chrome, Firefox, or Edge).

5.2.2 Accessing the Web App

After navigating to the local URL, you will see the web interface similar to that shown in Figure 3 of the Python Pip Package section.

5.2.3 Testing the Chatbot

To test the chatbot, enter a query such as "What's the weather in New York?" in the text box and click on the "Submit" button. The process might take some time depending on your machine. On a developer machine with 16 cores (Ryzen 7) and 16 GB RAM, it takes around 100 seconds.

While processing, you can check the terminal for the backend running LangChain, similar to what is shown in Figure 4 of the Python Pip Package section.

After some time, the response will be generated and displayed on the user interface, as shown in Figure 5 of the Python Pip Package section.

5.2.4 Shutting Down the Chatbot

To shut down the running chatbot, simply press the "Shutdown" button on the web interface. This will close the running bot at the backend.

Make sure to close the web browser window after shutting down the chatbot to free up system resources.

This concludes the first-time user experience for the Docker Package.

6 Troubleshooting

If you encounter any issues during installation or usage, refer to the following common troubleshooting steps:

- Ensure Docker is installed and running for Docker setup.
- Verify Python and pip installations for Python package setup.
- Check for sufficient system resources as per the requirements table.

7 Limitations

1. **Weather Forecast:** Forecast is currently restricted to two consecutive days from today. Queries for specific dates are not possible.
2. **One City Only:** Queries related to multiple cities simultaneously will result in a "Runtime Error".
3. **Computation Time:** The web app is executable on minimum requirements, but computation time can exceed 300 seconds for a simple weather query.

8 Online Extension

An interactive demonstration of this weather chatbot is hosted on Hugging Face Spaces, allowing users to experience its real-time capabilities: <https://huggingface.co/spaces/VatsalPatel18/weather-chatbot-phi3>

Note: Please allow at least 150 seconds for execution on an idle machine.

9 Computational Resources

- Source Code Repository: <https://github.com/VatsalPatel18/Weather-Chatbot-phi3>
- Docker Image: <https://hub.docker.com/r/vatsalpatel18/weather-chatbot-phi3>
- PyPI Package: <https://pypi.org/project/weather-chatbot-phi3/>