WeatherBot – A Chatbot weather Assistant 🌡️

🌞

A chatbot for real-time weather updates using Rasa

Repository:
https://github.com/katerinamas/-WeatherChatbot.git

Katerina Mastrofoti

Introduction

- **What is a Weather Bot?**

  - A chatbot designed to provide weather updates.

  - Fetches real-time weather data based on user input.

- **Why Build a Weather Bot?**

  - Helps users check temperature, weather conditions, and sunrise time easily.

  - Demonstrates chatbot capabilities using NLP and API integration.

⛅

Theoretical Background

- **What is a Chatbot?**

🤖

- A chatbot is an AI-powered program that simulates human conversation.
- Uses Natural Language Processing (NLP) to understand and respond to users.
- Can be rule-based (predefined responses) or AI-based (learning from interactions).

Understanding the Rasa Framework

- **What is Rasa?**

Open-source framework for building AI chatbots
Supports NLU (Understanding) & Dialogue Management
Can integrate with APIs & handle complex conversations

- **Why Use Rasa?**

Fully customizable & open-source
Understands context & handles **complex dialogues**
Easy API integration for dynamic responses

Understanding the Rasa Framework

- **Key Components**

Rasa NLU → Understands user input (intents & entities)
Rasa Core → Manages conversation flow & decisions

- **Key Concepts**

Intents: User intentions (e.g., ask_weather)

Entities: Extract data (e.g., city: Paris)

Slots: Stores extracted data for conversation memory
Responses: Predefined bot replies

Actions: Executes custom logic (e.g., calling APIs)

Stories & Rules: Define chatbot conversation flows

How Does the Weather Bot Work?

☔

- 1. User asks a weather-related question.
- 2. The bot classifies the user’s intent.
- 3. Extracts relevant entities like city name.
- 4. Fetches weather data from an external API.
- 5. Responds with formatted weather information.

User: *"What’s the temperature in Paris?"*
Rasa NLU → Identifies intent & extracts city: Paris
Rasa Core → Calls action_get_temperature
Action → Fetches data from API
Bot Response → *"The temperature in Paris is 15°C."*

☁️ API Integration ☁️

- Weather API Used: OpenWeatherMap API

How It Works:

- - User provides city name.
- - Chatbot sends a request to the API.
- - API returns temperature, condition, sunrise time.
- - Bot processes and presents the information.

Dialog Policy Experimentation

Memoization Policy:

- **Remembers frequently occurring conversation patterns.**

TED Policy:

- **Learns from past dialogues to predict responses.**

Rule Policy:

- **Handles structured responses (greetings, fallback).**

Why these choices?

- **Balances learning from conversations with predefined rules.**
- **Ensures bot can handle both structured and free-flowing interactions.**

Interaction Scenarios

NLU: Understanding User Input

Intents → Define user goals (ask_temperature, ask_weather_situation)

Entities → Extract structured data (e.g., city: Paris)

Example Utterances → Train chatbot to recognize user queries

Example:

User: *"What's the temperature in New York?"*

Intent: ask_temperature

Entity Extracted: city: New York

🌦️ Interaction Scenarios

Stories: Defining Conversation Flow

Stories → Predefined dialogue paths for different scenarios

Example Stories:
User: *"What's the weather in Berlin?"*
Intent: ask_weather_situation
Bot Calls Action: action_get_weather_situation
Bot Responds with API Data

Domain: Defining Bot Capabilities

Actions: Custom logic (action_get_temperature)

Responses: Predefined messages (utter_ask_city)

Slots: Store user data (e.g., city)

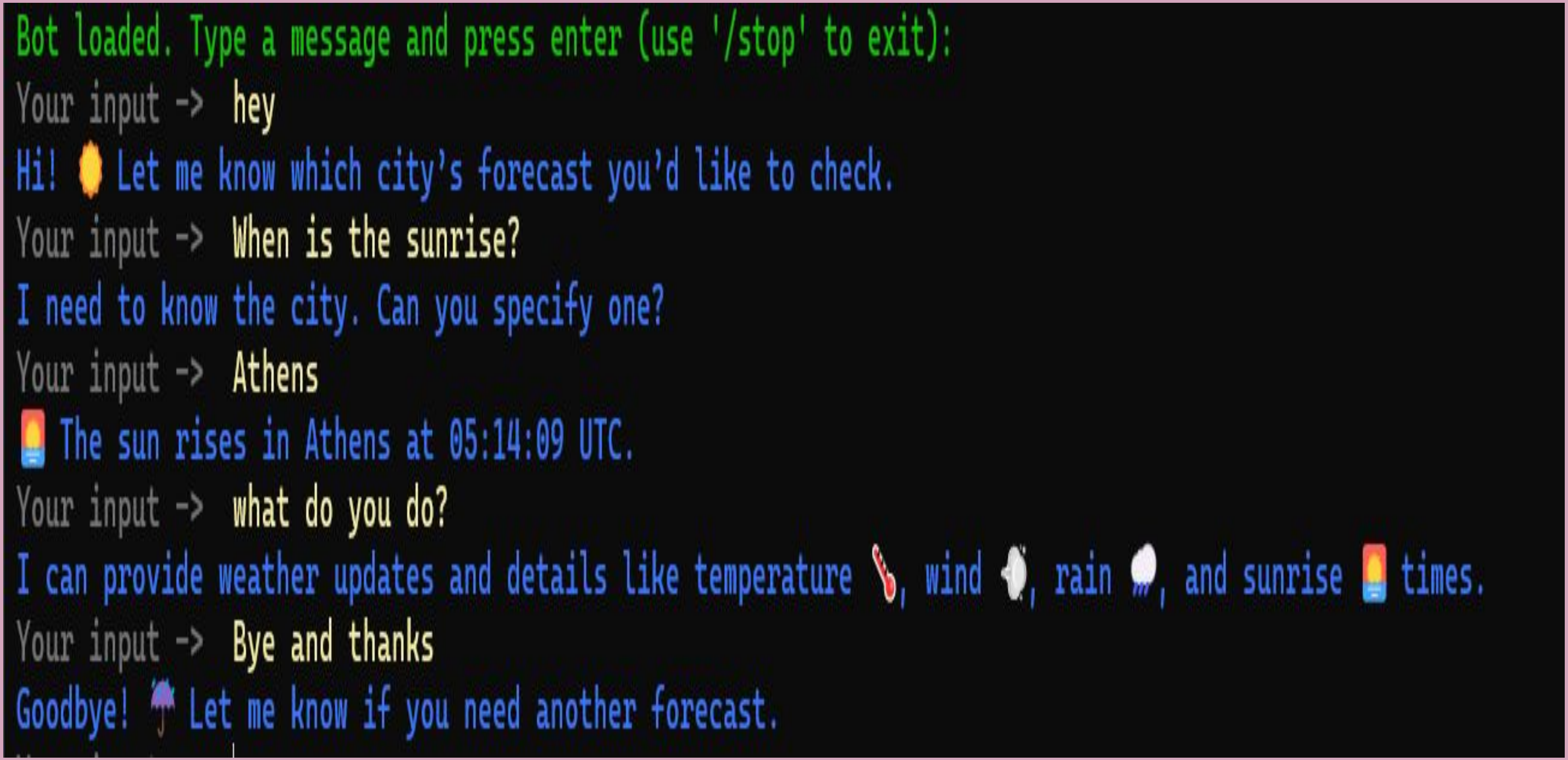
Handling Edge Cases

Missing Information? → utter_ask_city prompts user

Off-topic Queries? → utter_irrelevant politely declines

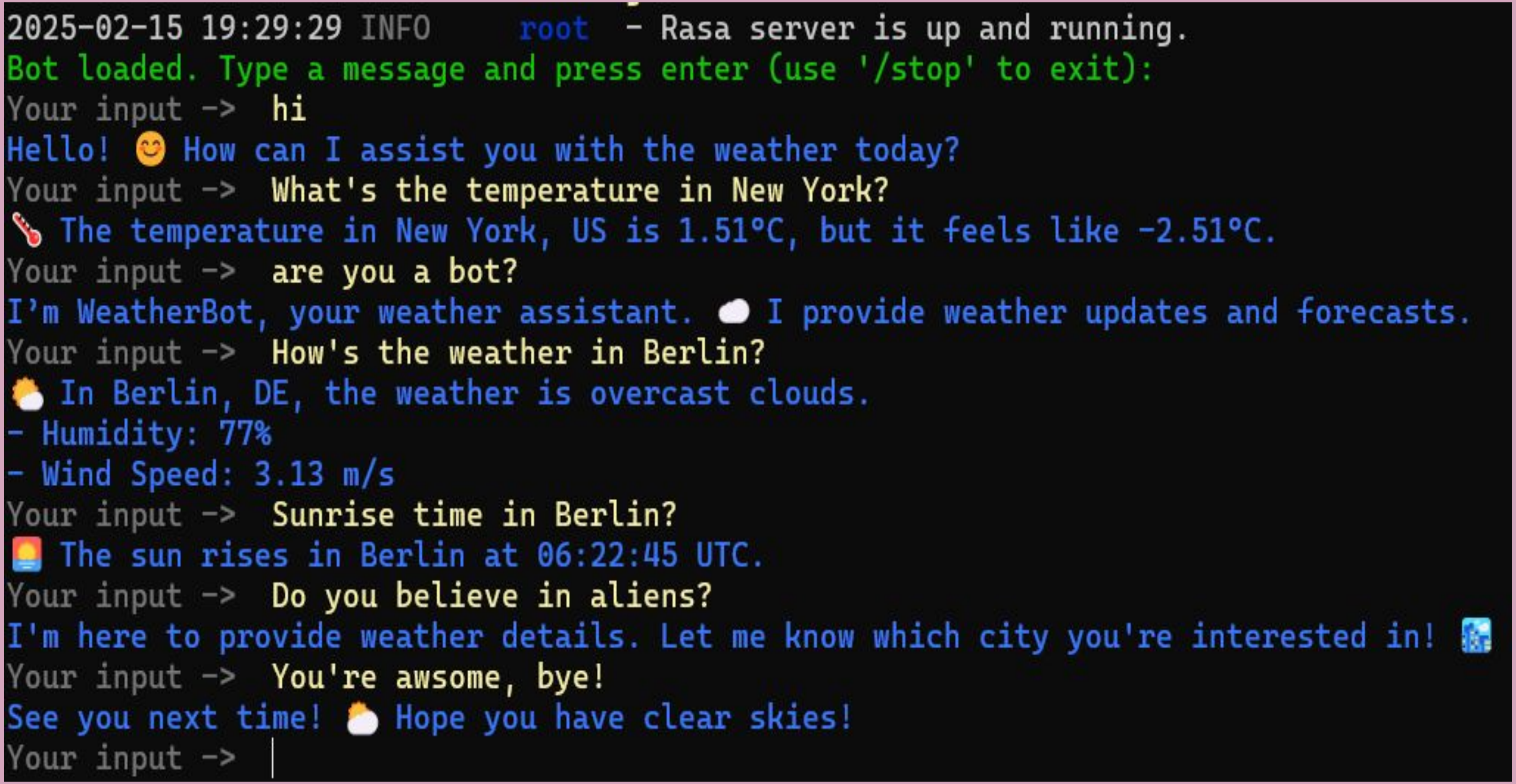
FAQs? → utter_capabilities explains bot functionalities

Example Interactions



```
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input ->  hey
Hi! 🌞 Let me know which city's forecast you'd like to check.
Your input ->  When is the sunrise?
I need to know the city. Can you specify one?
Your input ->  Athens
🌅 The sun rises in Athens at 05:14:09 UTC.
Your input ->  what do you do?
I can provide weather updates and details like temperature 🌡️, wind 🌬️, rain 🌧️, and sunrise 🌅 times.
Your input ->  Bye and thanks
Goodbye! ☔ Let me know if you need another forecast.
```

Example Interactions



```
2025-02-15 19:29:29 INFO     root  - Rasa server is up and running.
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input ->  hi
Hello! 😊 How can I assist you with the weather today?
Your input ->  What's the temperature in New York?
🌡️ The temperature in New York, US is 1.51°C, but it feels like -2.51°C.
Your input ->  are you a bot?
I’m WeatherBot, your weather assistant. ☁️ I provide weather updates and forecasts.
Your input ->  How's the weather in Berlin?
⛅ In Berlin, DE, the weather is overcast clouds.
- Humidity: 77%
- Wind Speed: 3.13 m/s
Your input ->  Sunrise time in Berlin?
🌅 The sun rises in Berlin at 06:22:45 UTC.
Your input ->  Do you believe in aliens?
I'm here to provide weather details. Let me know which city you're interested in! 🏙️
Your input ->  You're awsome, bye!
See you next time! ⛅ Hope you have clear skies!
Your input ->
```

Future Improvements

🔮

- **Expand Training Data:** Improve **intent recognition** with more diverse user queries
- **More Weather Insights: Provide forecasts, air quality index. Severe Weather Alerts:** Integrate **real-time weather warnings** for storms, heatwaves, and other extreme conditions. **Hourly Forecasts:** Allow users to check **temperature and precipitation trends** throughout the day.
- **Conversational Memory:** Allow the bot to **remember previous queries** for better follow-ups.
- Integrate **alternative APIs** to **increase accuracy** and provide more data.
- **Location-based Recommendations:** Use **GPS or saved locations** for automatic city recognition.
- **Chatbot Expansion Beyond Weather Road Condition Alerts:** Provide **real-time traffic/weather reports** for travelers. Suggest **best times to visit places** based on climate trends.

🌈 Conclusion

- The Weather Bot efficiently provides weather updates via conversation.
- Integrates NLU, API calls, and structured dialogue management.
- Future improvements will enhance accuracy and usability.

☀️