🌍 Travel Assistant Chatbot using Rasa

Subject: Special Topics in Language Technology: Multimodal and Dialogue Systems and Voice Assistants

Student: Nikos Vasilopoulos

Project Overview

Domain: Travel Assistance

- Destination recommendations
- Real-time weather information
- Local time across cities
- Upcoming events discovery

Why Travel Domain?

- Solves real problem: consolidates fragmented trip planning into one conversation
- Technical opportunity: multiple APIs, diverse intents, hybrid dialog policies
- Natural conversational interface for trip planning

System Architecture

Technologies:

✓ Rasa Open Source - Dialog management

✓ Rasa SDK - Custom actions

✓ Python 3 - Backend implementation

External Data Sources:

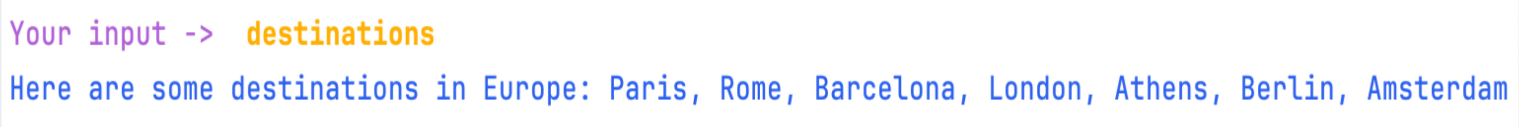
✓ OpenWeather API

✓ WorldTimeAPI

✓ Ticketmaster Discovery API

Scenario 1: Destination Recommendations

Functionality: User requests travel destination suggestions. The chatbot provides a predefined list of European capital cities.



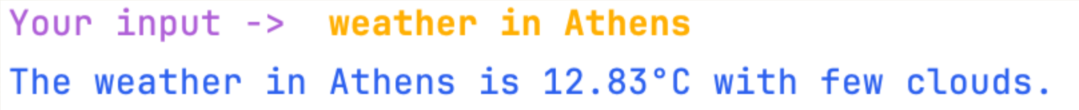
```
Your input ->  destinations
Here are some destinations in Europe: Paris, Rome, Barcelona, London, Athens, Berlin, Amsterdam
```

Technical Implementation

- Intent: ask_destinations
- Action: Custom action with static data
- Demonstrates: Basic intent recognition and response generation

Scenario 2: Weather Information

Functionality: User requests current weather for a specific city. The chatbot extracts the city name and retrieves real-time data.



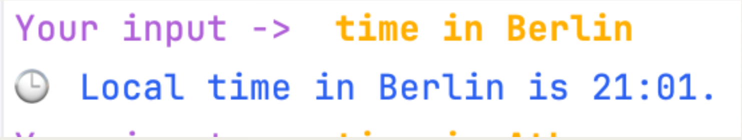
```
Your input ->  weather in Athens
The weather in Athens is 12.83°C with few clouds.
```

Technical Implementation

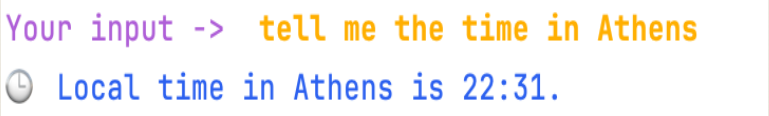
- Intent: ask_weather
- Entity: city (extracted from user message)
- API: OpenWeather API for real-time data
- Demonstrates: Entity extraction, slot filling, external API integration

Scenarios 3 & 4: Time and Events

Local Time Query

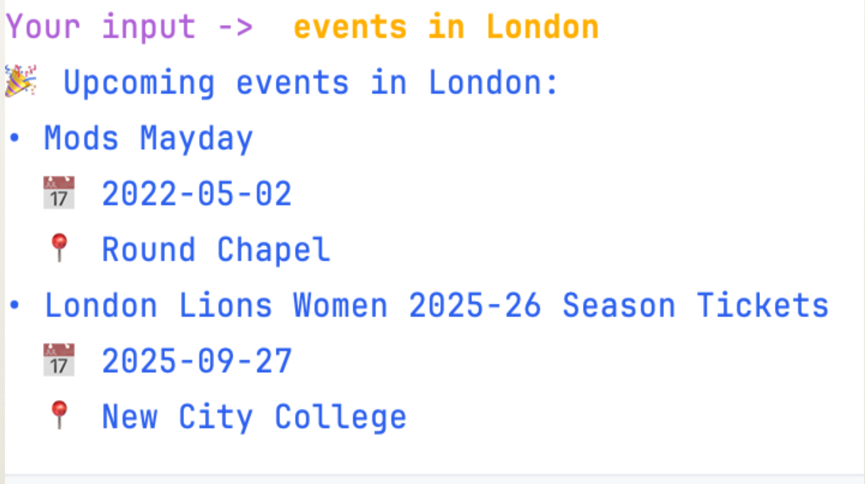


```
Your input ->  time in Berlin
🕒 Local time in Berlin is 21:01.
```



```
Your input ->  tell me the time in Athens
🕒 Local time in Athens is 22:31.
```

Events Discovery



```
Your input ->  events in London
🎉 Upcoming events in London:
• Mods Mayday
  📅 2022-05-02
  📍 Round Chapel
• London Lions Women 2025-26 Season Tickets
  📅 2025-09-27
  📍 New City College
```

Integrated Data Sources

🌤️ **OpenWeather API:**

Provides real-time weather conditions, temperature, and forecasts

🕐 **WorldTimeAPI:**

Returns accurate local time based on city time zones

🎫 **Ticketmaster Discovery API:**

Retrieves upcoming events filtered by city

➢ All APIs accessed via REST requests using Python *requests* library

Implementation Challenges

- **Challenge 1: City Name Ambiguity**

Multiple cities share the same name (e.g., Rome in Italy vs. Rome in USA)

Solution**:** Implemented city-to-country mapping to filter API results correctly

- **Challenge 2: Inconsistent API Coverage**

Some cities returned no events due to limited organizer availability

Solution: Added graceful fallback messages when no data is available

- **Challenge 3: Dialog Flow Conflicts**

Rules and stories conflicted, causing contradictory predictions

Solution: Clear separation between rule-based and story-based interactions

Error Handling

- **Network failures:** Try-except blocks with user-friendly messages
- **Missing API data:** Fallback responses instead of crashes
- **Invalid entities:** Clarification requests to users
- **API rate limits:** Graceful degradation with cached responses

Every error scenario provides meaningful feedback rather than technical error messages

Key Insights

- **Dialog Policy Balance:** Finding the right balance between rules and machine learning was crucial. Too many rules make the bot rigid, while pure ML can be inconsistent. The hybrid approach provides both predictability and conversational flexibility.
- **Real-World API Integration:** External APIs significantly enhance realism but add complexity. Error handling, data validation, and graceful fallbacks are essential for robust performance.
- **Entity Extraction Challenges:** Ambiguous entities (like duplicate city names) require careful handling through mapping strategies and context-aware filtering.

Limitations

- Limited to predefined cities major European destinations only)
- No multi-language support
- Time zone feature only works for cities in the destination list
- Event availability varies - not all cities have events data
- No all events included and they are generic, not filtered by category (music, sports, etc.)
- Command-line interface only

Future Improvements

- Expand city coverage globally
- Event filtering by category (music, sports, culture, etc.)
- Multi-language support
- User profiles and preferences
- Integration with booking systems (flights, hotels)
- Additional features for complete travel assistant (currency conversion, local tips, restaurant recommendations)

Conclusion

Project Achievements

✓ Four distinct interaction scenarios implemented

✓ Three real-world API integrations

✓ Hybrid dialog management approach

✓ Robust error handling mechanisms

✓ Complete documentation and setup guide