Foundation Models

Paolo Rota
paolo.rota@unitn.it

Marco Bronzini
marco.bronzini-1@unitn.it

University of Trento
November 12, 2025

Overview

Friday, October 10	Lab 1	• Metrics for Natural Language Generation (NLG) • Real-world use cases (MT, QA with RAG)
Wednesday, October 15	Lab 2	• Extract latent features from LLM embeddings by training a classification model
Friday, November 7	Lab 3	• Generate video captions using: a) CLIP-inspired models (**Co**ntrastive **Ca**ptioners) b) Vision-Language Models (**VLMs**)
Wednesday, November 12	Lab 4	• Agents using *LangChain*

Agents

An agent is a language model with *superpowers* 💪
It can be defined as a system that:

- Maps **inputs from the environment** to **actions**
- using some internal **policy** or **decision-making mechanism**.

Key Characteristics

1. **Autonomy:** It controls its own actions without direct human intervention;
2. **Perception:** It senses its environment through data inputs (e.g., history) or sensors.
3. **Action:** It influences the environment through output mechanisms;
4. **Goal-oriented behaviours:** It acts to achieve defined objectives;
5. **Adaptivity:** Many can learn from experience or adapt to changes in their environment.

Conversational Agents (i.e., Chat Bot)

A **conversational agent** is an agent designed to **interact with humans** through *natural language*:

a) Simulate conversation
b) Understand user's requests
c) Generate appropriate responses to achieve specific goals

A. Casheekar et al. Computer Science Review 52 (2024) 100632

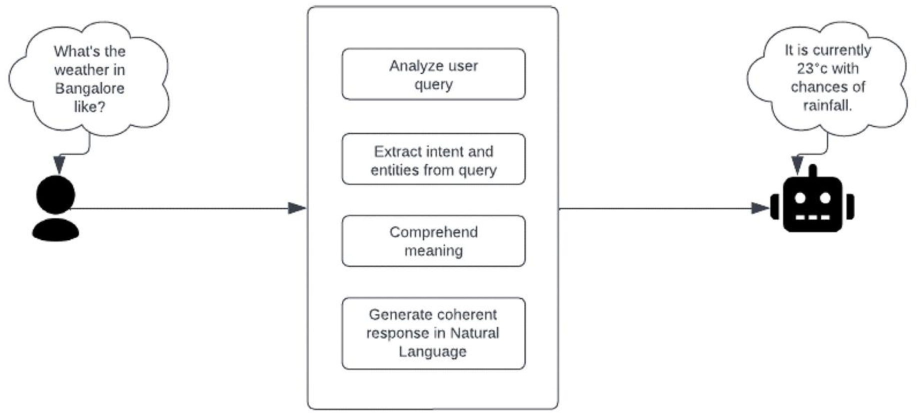
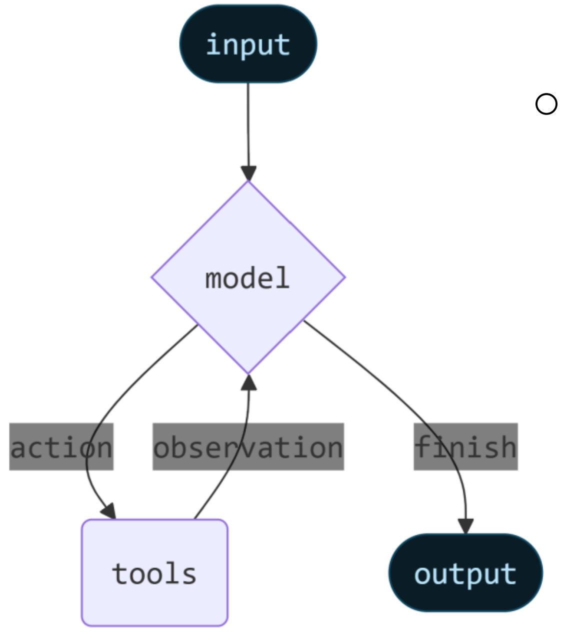

What's the weather in Bangalore like?
Analyze user query
Extract intent and entities from query
Comprehend meaning
Generate coherent response in Natural Language
It is currently 23°c with chances of rainfall.

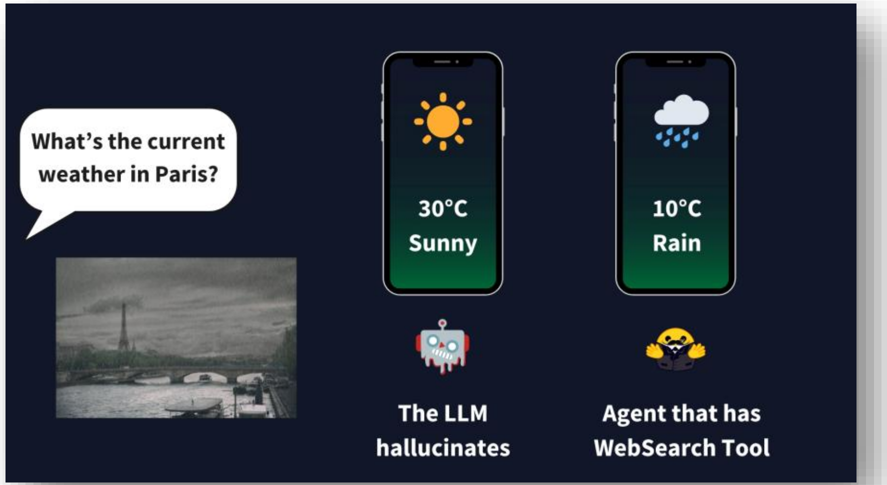
Fig. 1. AI-based chatbot – General interpretation.

Tools

- **Tools** refer to **external capabilities or resources** that:
  - an agent can invoke to perform tasks beyond its internal reasoning ability.
- It is typically a **specialized function** or external system that:
  - an agent uses whenever it decided the tool's capabilities are needed to reach its goal.



docs.langchain.com/oss/python/langchain/agents


What's the current weather in Paris?
30°C
Sunny
10°C
Rain
The LLM hallucinates
Agent that has WebSearch Tool

huggingface.co/learn/agents-course/en/unit1/to

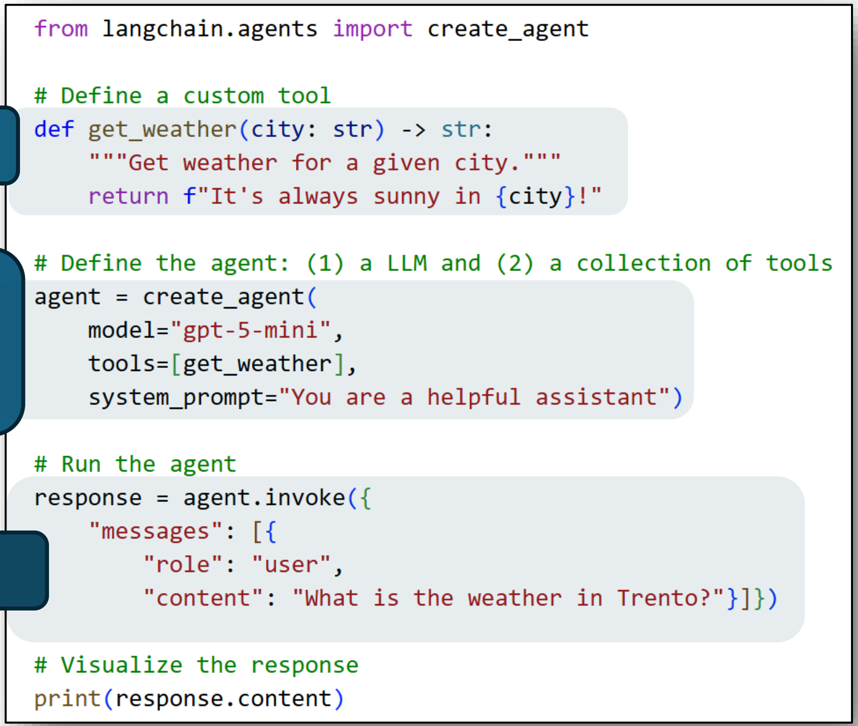
LangChain 🦜 🔗

- It's an open-source library enabling *building agents and applications powered by LLMs*.
- It provides a pre-built agent architecture and model integrations to help you get started quickly and seamlessly incorporate LLMs into your agents and applications

1 Define a TOOL

2 Create a tool-calling AGENT

3 Run the agent



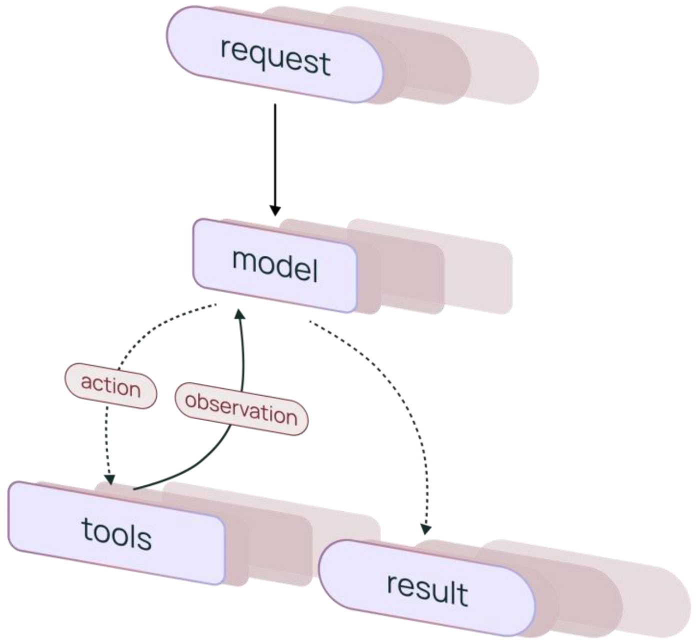
```python
from langchain.agents import create_agent

# Define a custom tool
def get_weather(city: str) -> str:
    """Get weather for a given city."""
    return f"It's always sunny in {city}!"

# Define the agent: (1) a LLM and (2) a collection of tools
agent = create_agent(
    model="gpt-5-mini",
    tools=[get_weather],
    system_prompt="You are a helpful assistant")

# Run the agent
response = agent.invoke({
    "messages": [{
        "role": "user",
        "content": "What is the weather in Trento?"}]})

# Visualize the response
print(response.content)
```



Practical Tutorial

Please, open the following notebook:
https://colab.research.google.com/github/saturnMars/FM_2025/blob/main/Lab4_agents.ipynb

QUIZ

MENTI (6672 8392)

https://www.menti.com/alna8gf2nrfw