
Learning LangGraph



.....

Course by **Matteo Falcioni**

Unibo, DIFA
Science City Lab

Mail: ***matteo.falcioni3@unibo.it***

GitHub: ***<https://github.com/MatteoFalcioni>***

.....

Table of Contents

First Part (fundamentals)

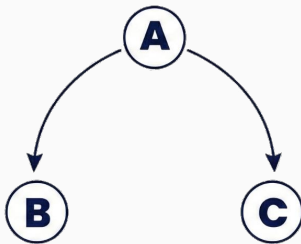
- 1. Introduction
- 2. A Simple Graph
- 3. LLMs & Agents in LangGraph
- 4. State Schema & Reducers
- 5. The Command Primitive
- 6. Agentic Graph, with memory
- 7. Agents Collaboration
- 8. Middleware: Llm summarization, human in the loop, todo list
- 9. Deep Agents

Second Part (advanced)

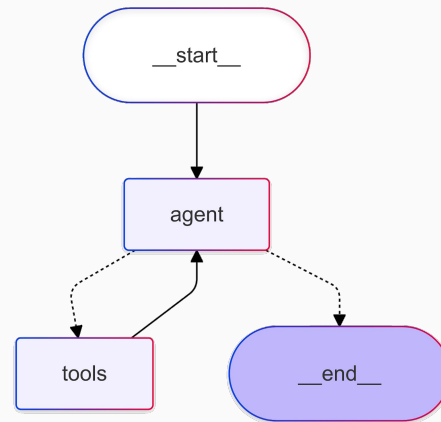
- 10. Supervised Graph
- 11. Hierarchical Agent Teams
- 12. **Project:** Multimodality - Vision Agents
- 13. **Project:** Multimodality - Voice Agents
- 14. **Project:** Towards *UrbIA* - complexifying our supervised graph
- 15. **Project:** RAG Agent
- 16. **Project:** *Your Suggestions*
- 17. Building a Production Ready Agent: *UrbIA* as an example of how to deploy and scale

LangChain & LangGraph

LangGraph is part of the LangChain ecosystem, but their structure is inherently different

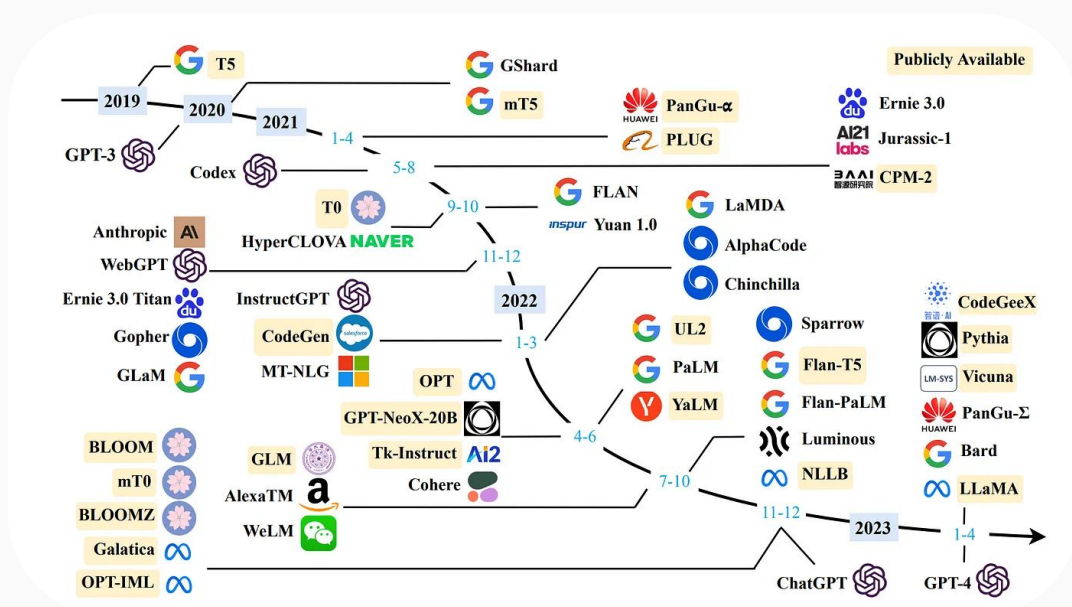
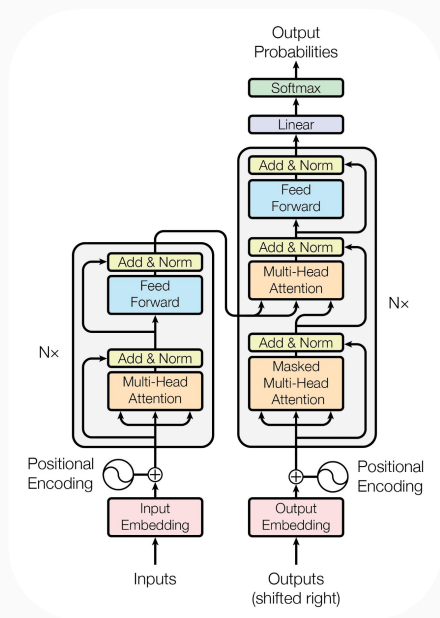


LangGraph allows us to have **loops** (→ agents with tools)



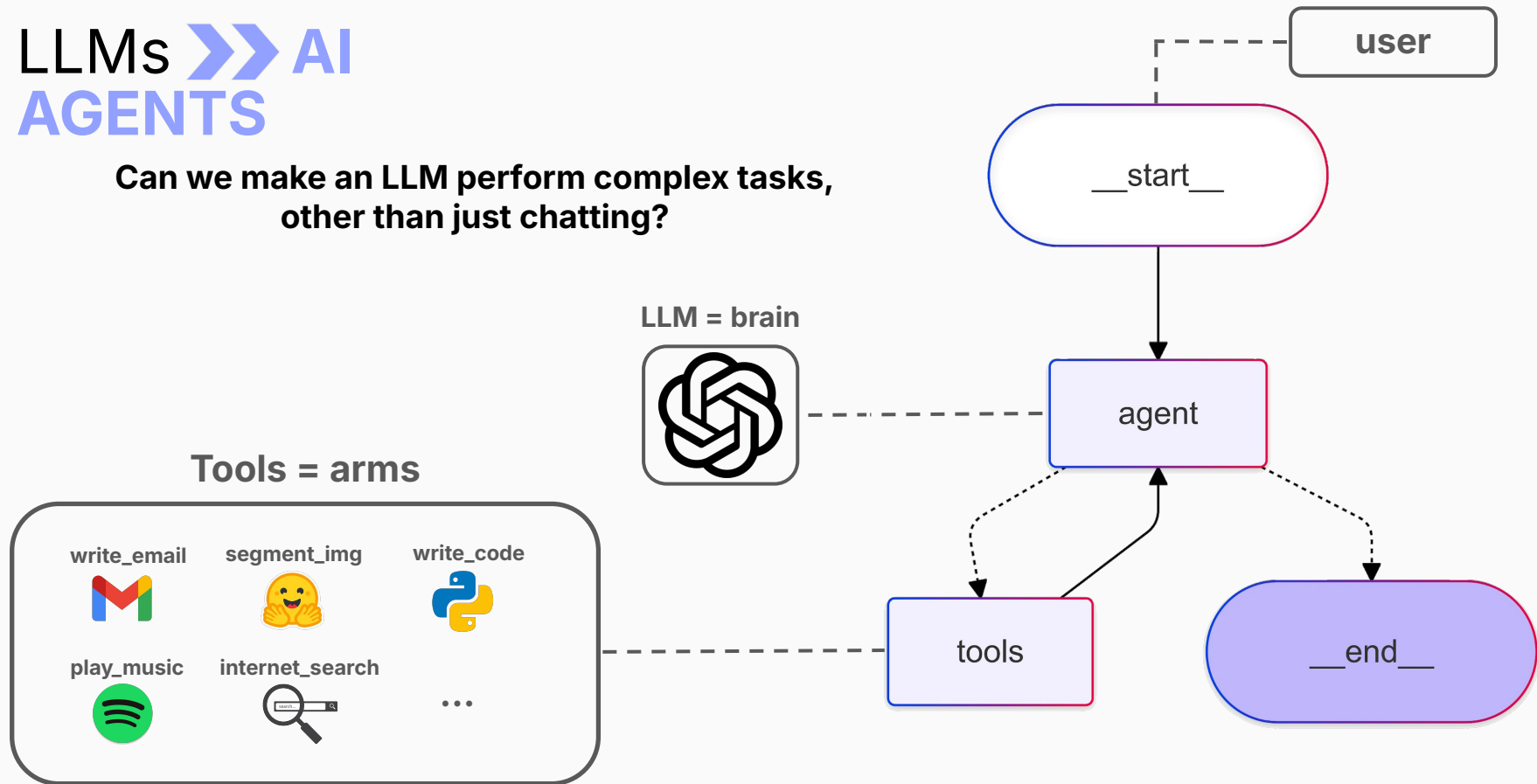
Large Language Models - LLMs

Attention is all you need! (2017)



LLMs >>> AI AGENTS

Can we make an LLM perform complex tasks, other than just chatting?



Tools Examples

Usual tool structure

```
59 @tool
60 def read_sources_tool(
61     runtime: ToolRuntime
62 ) -> Command:
63     """
64     Get the sources used in the analysis.
65     """
66     state = runtime.state
67     sources = state["sources"]
68     sources_str = "\n".join([f"- {source}" for source in sources])
69
70     return Command[tuple[()]](
71         update = {
72             "messages" : [ToolMessage(content=f"Sources: {sources_str}", tool_call_id=runtime.tool_call_id)],
73         }
74     )
```

Tools w/ API calls

```
1 from tavily import TavilyClient
2 import os
3 from langchain_core.tools import tool
4
5 # ----- Internet Search Tool -----
6
7 @tool
8 def internet_search(query):
9     """Search the internet for information"""
10     tavily_client = TavilyClient(api_key=os.environ["TAVILY_API_KEY"])
11     return tavily_client.search(query)
```

Resources

- Course Notebooks on GitHub: <https://github.com/MatteoFalcioni/Learning-LangGraph>
- LangChain Academy: [LangChain Academy](#)
- LangGraph Graph API: [Graph API overview - Docs by LangChain](#)
- LangChain Docs: [LangChain overview - Docs by LangChain](#)
- LangChain Reference: [Home | LangChain Reference](#)
- LangChain OpenTutorial:  [The LangChain Open Tutorial for Everyone](#)

For any information or doubts don't hesitate to contact me at matteo.falcioni3@unibo.it