AI Agents in LangGraph

link to course: https://learn.deeplearning.ai/courses/ai-agents-in-langgraph/lesson/1/undefined

Introduction

- welcome 😊
- most recurring patterns
  - planning
  - tool use (availability, how-to)
  - reflection (improving the results)
  - multi-agent communication
  - memory
- langchain is a framework that contains most of these patterns for agentic workflows
  - cyclic graphs
    - ReAct
    - self-refine
    - AlphaCodium
  - search: Tavily

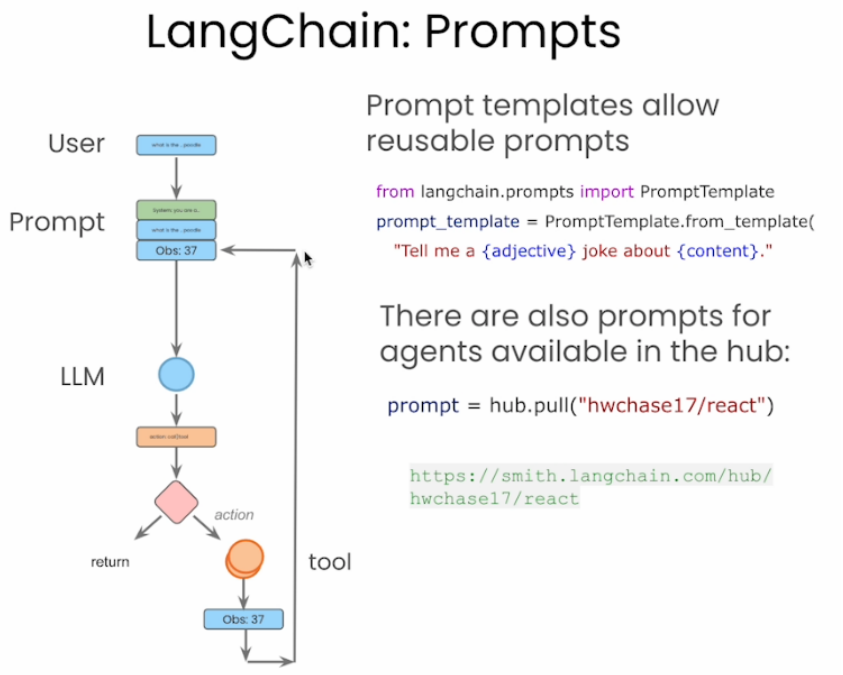
Build an Agent from Scratch

- we'll build an agent based on the ReAct pattern
  - thought → action → execution → observation → reasoning → `repeat`
- code comments
  - Agent class
    - take in a message from the user, a history of messages (ReAct loop)

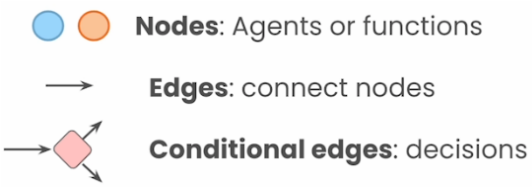
- ReAct agent
  - needs a very specific set of steps to follow

LangGraph Components

- break down: LangChain components
  - prompt templates



  - tools
    - describe and orchestrate the control flow for the LLM calls (cyclic graphs, persistence, human-in-the-loop)
  - graphs
    - nodes (agents or functions), edges (connections), conditional edges (decisions)



  - (agent) state
    - accessible from all over the graph
    - local to the graph
    - can be stored persistently
    - simple 🆚 complex

Agentic Search Tools

- why search tool?
  - RAG, context (sources)
- inside a search tool
  - query → sub-query
  - retrieve
    - chunked sources and top-k chunks
  - score & filtering
  - return top-k docs
- regular search tool 🆚 agentic one
  - agentic: focus on some structured (json mostly) and specific information

Persistence and Streaming

Human in the Loop

- state memory
  - StateSnapshot
  - get_state(thread)

Essay Writer

- our plan
  - plan the outline → research plan → generate [get documents, reflect, add documents, ...]
- implementation
  - our agents will be composed of a bunch of prompts for each type of task (plan, write, reflect, plan the research, critique)

LangChain Resources

- langchain doc
- langchain repo
  - tutos, cookbooks
- langgraph repo
- langsmith prompt hub

Conclusion

- multi-agent architecture

    - multiple agents work on the same shared state
- supervisor agent
    - coordination of multiple sub-agents
- flow engineering (AlphaCodium)
    - pipeline and loops
- plan & execute
- language agent tree search