

Summary and Highlights

Congratulations! You have completed this module. At this point, you know that:

- Reflection agents iteratively improve AI outputs by critically analyzing their performance through a feedback loop
- The generator produces content while the reflector provides critical feedback
- Prompt Engineering with LangChain guides LLMs in content generation and structured reflection using dynamic ChatPromptTemplates and message placeholders
- Agent state in LangGraph is defined using MessageGraph. It tracks conversation, accumulating messages and context across iterations
- Graph Construction involves defining nodes, connecting them with edges, setting an entry point, and using router nodes for dynamic decision-making and iterative loops
- Reflexion agents build on reflection agents by iteratively improving responses using self-critiques, external tools, and citations
- The reflection process involves a loop of generation, critique, and revision to enhance clarity, accuracy, and usefulness
- Reflexion agents can identify and fix their own weaknesses, improving with each cycle by analyzing prior outputs
- They can incorporate real-time data by calling external tools such as web search APIs, enhancing the relevance of responses
- Structured schema-based output helps agents distinguish between different components such as response, critique, and tool query
- The responder produces an object with fields such as query and response, which downstream components such as the revisor can build on
- The revisor refines the response by revising it, integrating tool outputs, and adding references to support the claims
- This entire process operates in an iterative cycle, with outputs and feedback passed through tools and stored in a response list across runs
- A search tool such as Tavily can be configured and invoked to enhance AI responses with external data
- Prompt engineering and schema design guide the LLM to produce structured reflections and focused answers