

Thêm Bộ Nhớ Ngắn Hạn vào Agent trong LangGraph

Trong bài học này, chúng ta sẽ tìm hiểu cách thêm **bộ nhớ ngắn hạn (short-term memory)** vào agent để nó có thể ghi nhớ ngữ cảnh của cuộc trò chuyện. Điều này giúp agent hiểu được các yêu cầu liên tiếp từ người dùng một cách chính xác hơn.

Vấn đề của Agent Không Có Bộ Nhớ

Trong bài tập trước, chúng ta đã xây dựng một agent đơn giản có thể thực hiện các phép tính toán học bằng cách di chuyển qua các node **"assistant"** và **"tools"**. Tuy nhiên, agent này có một hạn chế lớn: **nó không có bộ nhớ**.

Ví dụ minh họa:

- Người dùng yêu cầu:
"3 + 5."
→ Agent trả lời: "Tổng của 3 + 5 = 8"
- Người dùng tiếp tục:
"nhân kết quả với 2"

Nếu agent **không có bộ nhớ**, nó sẽ không hiểu "kết quả" ám chỉ kết quả trước đó và yêu cầu người dùng cung cấp lại số:

"Tôi cần một số cụ thể để nhân với hai. Bạn vui lòng cung cấp số bạn muốn nhân"

Điều này gây bất tiện vì người dùng phải lặp lại thông tin đã cung cấp trước đó.

Cách Thêm Bộ Nhớ Ngắn Hạn vào Agent

Trong **LangGraph**, việc thêm bộ nhớ ngắn hạn rất đơn giản bằng cách sử dụng module **MemorySaver**. Module này giúp lưu trữ thông tin tạm thời trong một phiên hội thoại cụ thể.

1. Sử dụng **MemorySaver** tại bước **compile**

Khi khởi tạo agent, chúng ta thêm bộ nhớ vào trong quá trình **compile**:

```
from langgraph.checkpoint.memory import MemorySaver

memory = MemorySaver()
```

```
react_graph_memory = builder.compile(checkpointer=memory)
```

2. Sử dụng `thread_id` để phân biệt các cuộc hội thoại

Để đảm bảo bộ nhớ chỉ áp dụng cho **một phiên trò chuyện cụ thể**, ta sử dụng `thread_id` khi gọi `invoke`:

```
config = {"configurable": {"thread_id": "1"}}

messages = [HumanMessage(content="Add 3 and 4.")]

messages = react_graph_memory.invoke({"messages": messages}, config)
```

Tác giả: Đỗ Ngọc Tú
Công Ty Phần Mềm VHTSoft

Phiên bản #1

Được tạo 21 tháng 4 2025 02:07:08 bởi Đỗ Ngọc Tú

Được cập nhật 21 tháng 4 2025 05:30:52 bởi Đỗ Ngọc Tú