

Tóm tắt chương

Tạm dừng một chút để cùng nhau ôn lại những gì chúng ta đã học được nhé!

Chúng ta bắt đầu chương này với cảm giác khá lo lắng – có lẽ bạn cũng từng nghĩ: "Cái LangGraph này nghe thật rối rắm và phức tạp", đúng không? Những khái niệm nghe có vẻ như dành riêng cho những người có siêu năng lực, phải có bằng tiến sĩ hay ít nhất là thạc sĩ mới hiểu nổi cách dùng nó!

Nhưng rồi khi đi đến cuối chương, ta nhận ra rằng **ai cũng có thể xây dựng ứng dụng với LangGraph**. Đằng sau những thuật ngữ nghe phức tạp đó thật ra lại là những khái niệm rất đơn giản. Và chúng ta đã thấy rằng **bất kỳ ai cũng có thể tạo ra những agent thông minh, thậm chí là những ứng dụng có thể thay đổi cả thế giới**.

Vậy cụ thể, ta đã làm gì?

- Đầu tiên, chúng ta học các **khái niệm cơ bản** của một ứng dụng LangGraph: state là gì, schema của state là gì, node là gì, edge là gì, quá trình compile là gì...
- Sau đó, ta xây dựng một app đơn giản dự đoán người dùng sẽ chọn Trà hay Coffee.
- Ta học được quy trình 5 bước để xây dựng một ứng dụng LangGraph:
 1. Định nghĩa state.
 2. Thêm node và edge.
 3. Biên dịch (compile).
 4. Hiển thị sơ đồ (visualize).
 5. Chạy ứng dụng.

Đơn giản đúng không nào?

- Tiếp theo, ta xây dựng một **chatbot** sử dụng LangGraph, học về khái niệm **reducer** – một từ nghe "kêu" nhưng thực chất chỉ là cách để cập nhật lại state theo cách riêng của mình.
- Ta sử dụng các **module có sẵn** như `message state`, giúp giảm lượng code cần viết.
- Sau đó, ta nâng cấp chatbot bằng cách **thêm tool** và **xây dựng hành vi agent**, nơi mà ứng dụng quyết định dùng chatbot hay gọi tool để trả lời người dùng.
- Ta học cách dùng `tool node` và `tool condition`, tạo router để định tuyến theo điều kiện đầu vào.

Và cuối cùng, **ta đã tạo ra agent đầu tiên của mình** bằng cách cho phép tool phản hồi trở lại LLM – chính là mô hình **ReAct**: act (gọi tool), observe (quan sát kết quả), reason (suy luận xem làm gì tiếp theo).

Sắp tới là gì?

Trong chương tiếp theo, chúng ta sẽ khám phá một chủ đề cực kỳ quan trọng trong thế giới LangGraph: **Memory (Bộ nhớ)**.

Dù trước giờ ta đã nói về state và một chút về memory, nhưng bạn sẽ thấy rằng, trong LangGraph, **memory không đơn giản chỉ là lưu trữ thông tin - nó là cốt lõi để tạo ra những agent thông minh và có tính phản hồi cao.**

Hãy chuẩn bị tinh thần để bước vào một chương siêu hấp dẫn nhé!

Tác giả: Đỗ Ngọc Tú
Công Ty Phần Mềm VHTSoft

Phiên bản #1

Được tạo 20 tháng 4 2025 04:02:20 bởi Đỗ Ngọc Tú

Được cập nhật 22 tháng 4 2025 02:13:50 bởi Đỗ Ngọc Tú