

Bộ nhớ cho AI agent Kiến trúc năm lớp

Vì sao agent quên sạch sau mỗi phiên, năm lớp bộ nhớ cần có, lộ trình dựng trong một tuần và năm phép thử để biết bộ nhớ chạy thật.

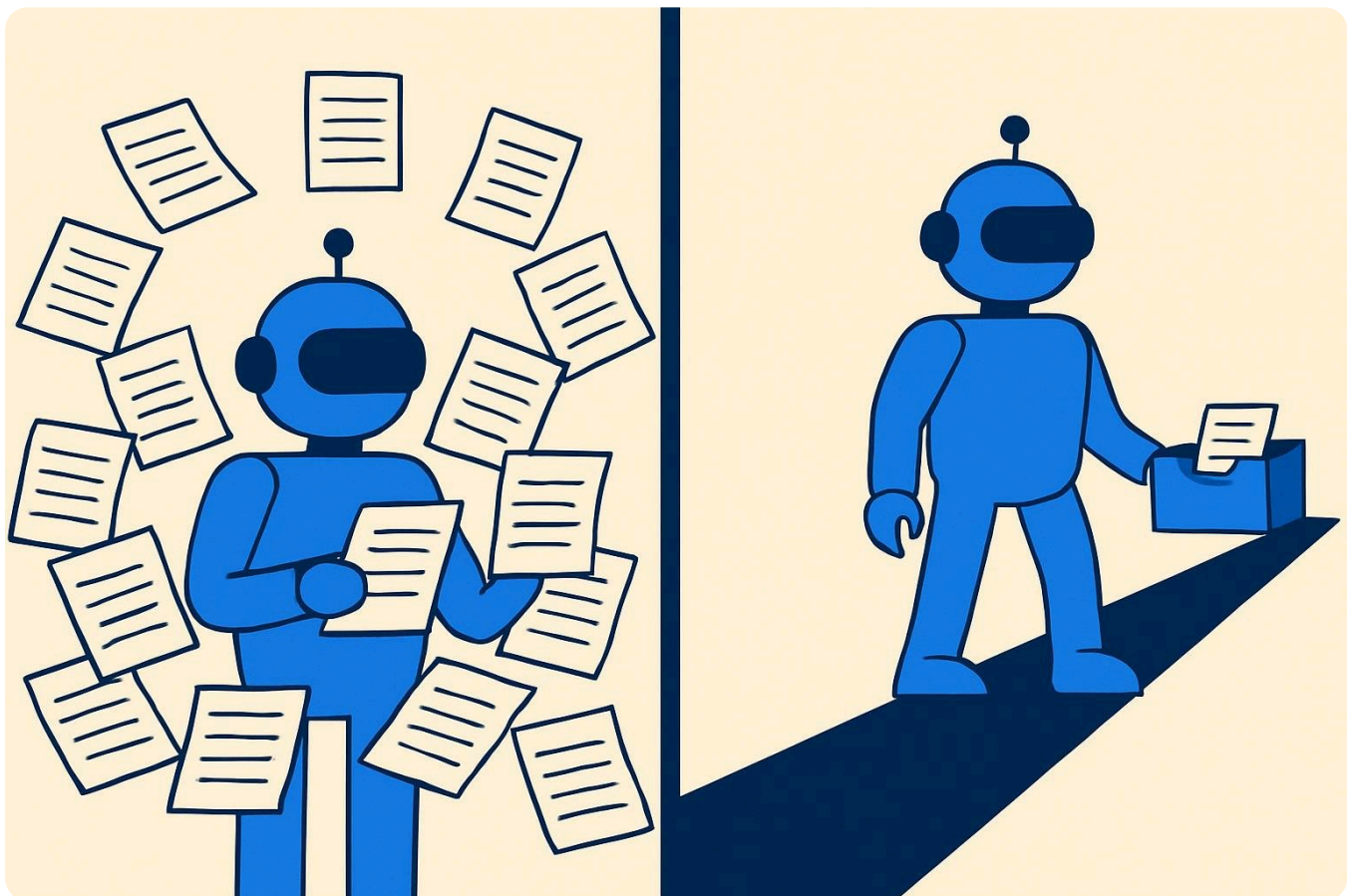
Mỗi sáng, agent của bạn mở ra với đầu óc trống rỗng. Những gì nó khám phá hôm qua, những chỉnh sửa bạn dặn, những lần nó đi vào ngõ cụt rồi tìm được lối ra — tất cả biến mất. Nó đọc lại đúng những tệp cũ, thử lại đúng những cách cũ, và tính tiền bạn đúng số token cũ.

Mô hình ngày càng thông minh hơn, nhưng một đồng nghiệp giỏi mà sáng nào cũng quên hết thì vẫn là người phải giao việc lại từ đầu mỗi ngày. Vấn đề ở đây không phải trí thông minh, mà là **bộ nhớ**. Bài này trình bày kiến trúc bộ nhớ năm lớp cho AI agent: mỗi lớp lưu gì, lấy ra khi nào, hết hạn ra sao, và triển khai theo thứ tự nào.

Vì sao agent quên

Mỗi lần gọi mô hình là một lần bắt đầu lại. Thứ duy nhất agent "nhớ" là những gì bạn nhét vào cửa sổ ngữ cảnh ngay lúc đó. Khi cửa sổ đầy, phần lớn hệ thống cắt bỏ từ đầu cuộc trò chuyện — tức là cắt đúng phần định nghĩa nhiệm vụ và những quyết định ban đầu, giữ lại phần tán gẫu gần nhất.

Đây là lý do nhiều agent làm tốt việc ngắn nhưng hỏng ở việc dài nhiều bước: đến bước thứ mười lăm, nó đã quên điều phát hiện ở bước thứ ba.



Không có bộ nhớ, mỗi phiên là một lần bắt đầu lại từ con số không.

Cách nhìn đúng

Cửa sổ ngữ cảnh lớn hơn không phải là bộ nhớ. Nó chỉ là bàn làm việc rộng hơn. Bộ nhớ là hệ thống quyết định cái gì được giữ lại sau khi dọn bàn.

Năm lớp bộ nhớ

Khung phân loại này đến từ CoALA (Cognitive Architectures for Language Agents), một khung học thuật mô tả agent ngôn ngữ theo kiểu kiến trúc nhận thức. Nó không phải phép ẩn dụ tâm lý học: mỗi lớp có cách lưu, cách truy xuất và vòng đời riêng.

LỚP	LƯU GÌ	LẤY RA KHI NÀO	HẾT HẠN
Bộ nhớ làm việc	Ngữ cảnh hiện tại	Luôn nạp	Hết phiên
Bộ nhớ tình tiết	Đã xảy ra chuyện gì	Khi gặp việc tương tự	30–90 ngày
Bộ nhớ ngữ nghĩa	Điều gì là đúng	Khi cần dữ kiện	Khi bị thay thế
Bộ nhớ quy trình	Làm việc đó thế nào	Khi khóp tình huống	Khi cách làm đổi
Cơ chế quên	Cần xóa gì	Chạy theo lịch	Liên tục

Người dùng → **Lớp 1 · Bộ nhớ làm việc** → (ngữ cảnh đầy) → **Lớp 2 · Tình tiết** → (chưng cất) → **Lớp 3 · Ngữ nghĩa** → (cách làm lặp lại) → **Lớp 4 · Quy trình**

Chiều ngược lại: ba lớp dưới nạp phần liên quan lên bộ nhớ làm việc mỗi khi có việc mới. **Lớp 5 · Cơ chế quên** chạy nền trên cả ba lớp lưu lâu dài.

Lớp 1 — Bộ nhớ làm việc

Chính là cửa sổ ngữ cảnh: chỉ dẫn hệ thống, cuộc hội thoại hiện tại, kết quả công cụ, tài liệu vừa tra cứu. Điểm cần sửa nằm ở lúc nó đầy. Thay vì cắt từ đầu, hãy trích ra những quyết định và dữ kiện quan trọng, đẩy xuống lớp bên dưới, rồi mới nén phần còn lại.

Lớp 2 — Bộ nhớ tình tiết

Nhật ký trải nghiệm: đã thử việc gì, cách nào chạy được, cách nào hỏng và hỏng vì sao. Trước khi bắt tay vào việc mới, agent tra xem đã từng gặp việc tương tự chưa. Một lần tra cứu như vậy có thể tiết kiệm cả buổi dò lại lỗi cũ.

Nên lưu: mô tả nhiệm vụ, kết quả, lỗi gặp phải và cách xử lý, quyết định kèm lý do, góp ý của người dùng. Không nên lưu: toàn bộ chuỗi suy luận dài dòng, nội dung tài liệu thô, các bản nháp trung gian.

Lớp 3 — Bộ nhớ ngữ nghĩa

Những dữ kiện bên: khách hàng này thuộc nhóm nào, hệ thống đặt ở đâu, quy định công ty ra sao. Chúng không hết hạn khi đóng phiên làm việc.

Điểm mấu chốt là **phải có lược đồ**. Khi định nghĩa rõ có những loại thực thể nào, quan hệ nào hợp lệ, trường nào bắt buộc, kho dữ kiện mới không biến thành bãi rác ghi chép. Và khi một dữ kiện đổi, phải thay thế bản cũ chứ không ghi thêm bản mới bên cạnh — nếu không, agent sẽ có hai sự thật trái nhau và dùng cái nào tùy may rủi.

Lớp 4 — Bộ nhớ quy trình

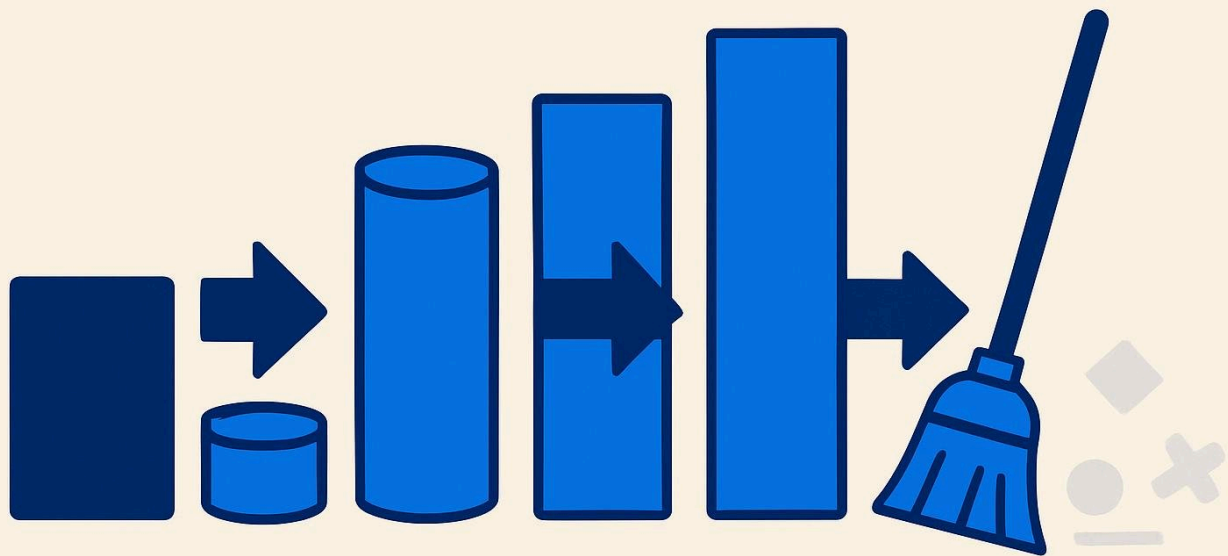
Cách làm đã thành công, lưu lại thành kỹ năng tái dùng: tên, điều kiện kích hoạt, các bước, công cụ cần, tiêu chí thành công. Lần sau gặp đúng loại việc, agent nạp kỹ năng và làm theo thay vì nghĩ lại từ đầu.

Đừng biến mọi thành công thành kỹ năng. Chỉ thăng cấp khi việc đó lặp lại đều, đã thành công vài lần, và các bước mô tả được rõ ràng. Một lần may mắn là tình tiết, không phải kỹ năng. Và kỹ năng cần đánh phiên bản: khi quy trình thật đổi, kỹ năng cũ trở thành sai — một kỹ năng lỗi thời còn nguy hiểm hơn không có kỹ năng, vì agent làm sai một cách rất tự tin.

Lớp 5 — Cơ chế quên

Lớp ít ai làm nhất và cũng là lớp giữ cho bốn lớp kia còn dùng được. Agent không bao giờ quên sẽ tích tụ mâu thuẫn: khách đã đổi số điện thoại nhưng số cũ vẫn nằm đó, chính sách đã thay nhưng bản cũ vẫn được trích dẫn.

Ba việc cơ chế quên phải làm: cho hết hạn những gì quá cũ, thay thế dữ kiện cũ bằng dữ kiện mới, và đánh dấu những mâu thuẫn thật sự để người xem xét.



Năm lớp nối thành một đường ống, cơ chế quên giữ cho cả chuỗi không tích mâu thuẫn.

Lộ trình một tuần

Không cần dừng cả năm lớp cùng lúc. Thứ tự dưới đây đi từ lớp dễ nhất giải quyết nỗi đau lớn nhất.

1 Ngày 1–2: Bộ nhớ tình tiết

Ghi kết quả mỗi nhiệm vụ vào một tệp. Trước việc mới, đọc vài mục gần nhất và tóm tắt vào ngữ cảnh.

2 Ngày 3–4: Bộ nhớ ngữ nghĩa

Trích các dữ kiện bền từ nhật ký: sở thích khách hàng, cấu hình dự án, quy ước nội bộ. Lưu vào bảng đơn giản, nạp đúng phần liên quan.

3 Ngày 5–6: Kỹ năng và quên

Việc nào lặp lại thành công ba lần thì lưu thành kỹ năng. Đặt hạn cho tình tiết cũ, dữ kiện mới thay dữ kiện cũ.

4 Ngày 7: Nối cả đường ống

Ngữ cảnh đầy thì đẩy xuống tình tiết, tình tiết lặp lại thì chưng cất thành dữ kiện, dữ kiện mô tả cách làm thì thành kỹ năng, cơ chế quên chạy hàng ngày.

Phép thử cuối cùng rất đơn giản: giao một việc kéo dài nhiều phiên. Nếu sang phiên sau agent vẫn nhớ quyết định của phiên trước mà không phải đọc lại toàn bộ tài liệu, đường ống đã chạy.

Khi nào không cần bộ nhớ

Không phải agent nào cũng cần. Câu hỏi để quyết: *nếu nhớ được lần trước, kết quả lần này có tốt hơn không?*

TÌNH HUỐNG	CẦN BỘ NHỚ	VÌ SAO
Định dạng lại một đoạn văn bản	Không	Việc một lần, không có gì để nhớ
Chăm sóc khách hàng	Có	Cần lịch sử và sở thích của khách
Báo cáo định kỳ cùng một nguồn dữ liệu	Có	Nhớ định dạng và quy ước đã chốt
Nghĩ ý tưởng sáng tạo	Không	Mới mẻ quan trọng hơn nhất quán
Sửa lỗi qua nhiều buổi	Có	Phải nhớ đã thử gì và hỏng ở đâu

Bộ nhớ cũng có giá của nó

Mỗi lần truy xuất thêm một khoảng chờ. Mỗi dữ kiện lưu lại tốn chỗ. Mỗi lần dọn dẹp tốn tính toán. Kiến trúc bộ nhớ phải tự chứng minh: phần tiết kiệm được nhờ không nạp lại mọi thứ phải lớn hơn chi phí duy trì nó.

Điều này chỉ đạt được khi lưu có kỷ luật — lưu quyết định và kết quả, không lưu mọi thứ. Lưu tất cả là cách chắc chắn nhất để tín hiệu chìm trong nhiễu và chi phí phình lên mà chất lượng không đổi.

Bộ nhớ không thay được suy luận. Lưu được dữ kiện "khách thích gói cơ bản" không có nghĩa agent biết vận dụng điều đó đúng lúc. Bộ nhớ làm cho thông tin sẵn có, phần còn lại vẫn là việc của mô hình và của thiết kế.

Năm phép thử để biết bộ nhớ có chạy

Quên Nói một dữ kiện ở phiên 1, hỏi lại ở phiên 2	Mâu thuẫn Nạp hai dữ kiện trái nhau, xem agent có phát hiện	Hết hạn Dữ kiện quá hạn có còn được dùng không
---	---	--

Thêm hai phép nữa: **thăng cấp kỹ năng** — làm cùng loại việc thành công ba lần, lần thứ tư agent phải dùng kỹ năng đã lưu thay vì nghĩ lại; và **tải nặng** — nhét hàng nghìn bản ghi rồi đo thời gian truy xuất. Hệ

thống chạy tốt với trăm bản ghi mà chậm ở chục nghìn thì chưa dùng được thật.

AIECOS áp dụng thế nào

Chatbot của chính AIECOS chạy trên mô hình này. Lịch sử hội thoại của từng khách được lưu riêng để agent nhớ mạch trao đổi — đó là lớp tình tiết ở mức cơ bản. Kho tài liệu giải pháp được cắt thành đoạn, vector hoá và lưu trong cơ sở dữ liệu, agent chỉ truy vấn đúng phần liên quan cho mỗi câu hỏi thay vì nạp toàn bộ — đó là lớp ngữ nghĩa, và cũng là lý do câu trả lời bám sát tài liệu thật.

Với doanh nghiệp, [hệ thống AI xây riêng](#) đặt bộ nhớ ở tầng dữ liệu chung: hồ sơ khách hàng hợp nhất, tài liệu nội bộ, lịch sử tương tác — để mọi agent cùng nhìn một nguồn. Khi có nhiều agent, phân phân vai và ranh giới dữ liệu đã nói trong bài [quản trị khi doanh nghiệp có nhiều AI agent](#). Và vì bộ nhớ chỉ tốt bằng dữ liệu nạp vào nó, nên đọc kèm bài [dọn dữ liệu trước khi dùng AI](#).

Câu hỏi thường gặp

Bộ nhớ khác gì RAG?

RAG là tra cứu tài liệu có sẵn. Bộ nhớ là những gì agent tự tích lũy trong lúc làm việc: đã thử gì, học được gì, cách nào chạy được. Một hệ thống đầy đủ cần cả hai.

Doanh nghiệp nhỏ có cần đủ năm lớp không?

Không. Bắt đầu bằng lớp tình tiết, thường đã gỡ được phần lớn nỗi đau. Thêm lớp khi thấy đúng triệu chứng: agent quên sở thích khách thì thêm ngữ nghĩa, agent làm lại từ đầu mỗi lần thì thêm quy trình.

Dữ liệu trong bộ nhớ có an toàn không?

Phụ thuộc cách phân quyền. Nguyên tắc nên theo: mỗi agent chỉ đọc phần dữ liệu thuộc phạm vi của nó, dữ liệu của khách hàng hay dự án này không rò sang dự án khác, và mọi thao tác đều ghi lại.

Nên tự xây hay dùng thư viện có sẵn?

Giai đoạn đầu nên tự xây bản đơn giản nhất để hiểu mình cần gì. Khi khối lượng lớn và cần tìm kiếm nhanh, chuyển sang các thư viện bộ nhớ chuyên dụng sẽ đỡ công vận hành.

[Tải PDF](#)

Bạn muốn agent của mình nhớ được việc hôm qua thay vì bắt đầu lại mỗi sáng? [Đặt lịch demo miễn phí](#), chúng tôi rà kiến trúc hiện tại và chỉ ra lớp nên làm trước.

Nguồn tham khảo

Bài viết tổng hợp từ khung CoALA (Cognitive Architectures for Language Agents) và cách làm của các hệ thống bộ nhớ cho agent như Mem0, Zep, cùng tài liệu kỹ thuật công khai về kiến trúc agent. Những con số hiệu năng mà các bên này công bố không được dẫn lại ở đây vì chúng tôi chưa kiểm chứng được nguồn gốc.

AIECOS — thiết kế và xây dựng hệ thống AI riêng cho doanh nghiệp. aiecos.ai · hello@aiecos.ai

Tài liệu do AIECOS biên soạn, không liên kết với bất kỳ nhà cung cấp nào được nhắc tới trong bài.