

Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

“Một mình làm chẳng nên non, ba người chụm lại thành hòn núi cao.”

— Tục ngữ Việt Nam

Sức mạnh của tập thể luôn lớn hơn cá nhân — hợp tác là chìa khóa để đạt những mục tiêu lớn.

TIN TỨC NỔI BẬT

1

Claude Code đấu với Cursor 2026: 80.8% SWE-bench, 1M Context [Đã kiểm chứng]

Claude Code vs Cursor 2026: 80.8% SWE-bench, 1M Context [Tested]

tech-insider.org [Đọc bài viết →](#)

Một thử nghiệm benchmark gần đây đã so sánh Claude Code với Cursor 2026, với kết quả đáng chú ý. Thử nghiệm, dường như tập trung vào SWE-bench, một thước đo hiệu suất của mô hình ngôn ngữ, đã cho kết quả điểm số 80,8% cho Claude Code. Ngoài ra, thử nghiệm đã sử dụng một ngữ cảnh 1 triệu, cung cấp một đánh giá toàn diện về khả năng của mô hình. SWE-bench là một chuẩn mực được công nhận rộng rãi để đánh giá hiệu suất của mô hình ngôn ngữ, khiến thử nghiệm này trở thành một chỉ số quan trọng về khả năng của Claude Code. Điểm số 80,8% cho thấy Claude Code đã thể hiện hiệu suất mạnh mẽ trong lĩnh vực này. Việc sử dụng ngữ cảnh 1 triệu thêm độ sâu hơn vào thử nghiệm, vì nó cho phép đánh giá tinh tế hơn về khả năng hiểu và tạo ra của mô hình. Mặc dù các chi tiết của thử nghiệm và phương pháp luận không được cung cấp, kết quả mang lại một cái nhìn về hiệu suất của Claude Code so với Cursor 2026. Phân tích và thử nghiệm thêm sẽ cần thiết để hiểu đầy đủ ý nghĩa của những kết quả này.

2

Tổng quan về Agent Frameworks: Lựa chọn Runtime phù hợp cho Vận hành AI Doanh nghiệp

State of Agent Frameworks: Choosing the Right Runtime for Enterprise AI Execution

Medium [Đọc bài viết →](#)

Bài viết "Trạng thái của các Framework Trình điều khiển: Chọn Thời gian chạy Đúng cho Thực thi AI Doanh nghiệp" khám phá cảnh quan hiện tại của các framework trình điều khiển trong bối cảnh thực thi AI doanh nghiệp. Các framework trình điều khiển là các nền tảng phần mềm cho phép phát triển và triển khai các model trí tuệ nhân tạo (AI) và học máy (ML). Bài viết nhấn mạnh tầm quan trọng của việc chọn thời gian chạy đúng cho thực thi AI doanh nghiệp, vì nó có thể ảnh hưởng đáng kể đến hiệu suất, khả năng mở rộng và độ tin cậy của các ứng dụng AI. Bài viết thảo luận về các framework trình điều khiển khác nhau, bao gồm Rasa, Dialogflow và Botpress, là những lựa chọn phổ biến để xây dựng các ứng dụng AI trò chuyện. Nó cũng đề cập đến tầm quan trọng của việc xem xét các yếu tố như dễ sử dụng, tùy chỉnh và tích hợp với các hệ thống hiện có khi chọn một framework trình điều khiển. Hơn nữa, bài viết nhấn mạnh nhu cầu về một môi trường thời gian chạy mạnh mẽ có thể xử lý các phức tạp của thực thi AI doanh nghiệp, bao gồm quản lý dữ liệu, bảo mật và khả năng mở rộng. Bằng cách hiểu trạng thái hiện tại của các framework trình điều khiển, các tổ chức có thể đưa ra quyết định thông minh về việc chọn thời gian chạy nào cho các sáng kiến AI của họ.

3

Vai trò thầm lặng của Toán học và Algorithms trong MCP & Multi-Agent Systems

The Silent Role of Mathematics and Algorithms in MCP & Multi-Agent Systems

Cisco Blogs [Đọc bài viết →](#)

Bài viết nhấn mạnh vai trò quan trọng nhưng thường bị bỏ qua của toán học và thuật toán trong Computation nhiều bên (MCP) và Hệ thống nhiều tác nhân. MCP cho phép tính toán bảo mật trên dữ liệu riêng tư mà không tiết lộ đầu vào cá nhân, trong khi Hệ thống nhiều tác nhân giúp thúc đẩy sự hợp tác và ra quyết định giữa các thực thể tự chủ. Toán học và thuật toán đóng vai trò im lặng nhưng quan trọng trong cả hai lĩnh vực. Trong MCP, các khái niệm toán học như mã hóa đồng hình và giao thức tính toán nhiều bên bảo mật đảm bảo tính toàn vẹn và bảo mật của dữ liệu. Các thuật toán như Chia sẻ Bí mật Shamir và Giao thức Yao cho phép tính toán bảo mật trên dữ liệu riêng tư, cho phép các bên hợp tác mà không ảnh hưởng đến đầu vào cá nhân của họ. Trong Hệ thống nhiều tác nhân, các mô hình toán học và thuật toán giúp thúc đẩy sự phối hợp và ra quyết định của các tác nhân tự chủ. Các kỹ thuật như lý thuyết trò chơi và lý thuyết đồ thị giúp các tác nhân điều hướng các kịch bản phức tạp và đưa ra quyết định tối

ưu. Bài viết nhấn mạnh tầm quan trọng của toán học và thuật toán trong các lĩnh vực này, thường bị bỏ qua để ủng hộ những tiến bộ công nghệ rõ ràng hơn. Bằng cách nhận ra tầm quan trọng của chúng, các nhà nghiên cứu và nhà phát triển có thể tạo ra các hệ thống mạnh mẽ và bảo mật hơn.

4

Agentic Batch Changes hiện đã có bản public beta

Agentic Batch Changes is now in public beta

Sourcegraph Blog [Đọc bài viết →](#)

Agentic Batch Changes, một công cụ được hỗ trợ bởi AI, đã bước vào giai đoạn beta công khai. Công nghệ này sử dụng một AI agent để quản lý việc di chuyển mã trên quy mô lớn trên nhiều kho lưu trữ. Trình đại lý này có khả năng xác định và thực hiện các thay đổi cần thiết, đảm bảo rằng tất cả các yêu cầu kéo (PRs) trở nên có thể hợp nhất. Quá trình này có thể được áp dụng cho hàng trăm kho lưu trữ, giúp đơn giản hóa quá trình di chuyển mã. Mục tiêu chính của Agentic Batch Changes là đơn giản hóa và tự động hóa nhiệm vụ di chuyển mã thường phức tạp và tốn thời gian. Bằng cách tận dụng AI, công cụ này nhằm giảm thiểu nỗ lực thủ công cần thiết để cập nhật cơ sở mã, giúp cho các nhà phát triển và nhóm làm việc hiệu quả hơn. Với công cụ hiện đang trong giai đoạn beta công khai, các nhà phát triển có thể thử nghiệm và trải nghiệm các khả năng của Agentic Batch Changes, cung cấp phản hồi quý giá để cải thiện thêm công nghệ.

5

Chính quyền Trump gỡ bỏ hạn chế xuất khẩu các AI Model Mythos và Fable của Anthropic

The Trump Administration Is Lifting Its Export Controls on Anthropic's Mythos and Fable AI Models

Wired [Đọc bài viết →](#)

Chính quyền Trump đã dỡ bỏ các hạn chế xuất khẩu đối với các mô hình AI mạnh mẽ của Anthropic, bao gồm Mythos 5 và Fable 5, sau khi công ty này đạt được thỏa thuận với Bộ Thương mại. Một bức thư từ Bộ trưởng Thương mại Howard Lutnick gửi đến đồng sáng lập Anthropic Tom Brown xác nhận rằng không cần giấy phép để xuất khẩu hoặc chuyển giao các mô hình này. Quyết định này được đưa ra sau khi Anthropic đồng ý tăng cường các biện pháp bảo vệ để ngăn chặn người dùng vượt qua các hạn chế an toàn, đặc biệt là những hạn chế liên quan đến an ninh mạng. Công ty đã cam kết chủ động phát hiện

và giải quyết các rủi ro bảo mật, hợp tác với chính phủ Mỹ về các giao thức và tiêu chuẩn, và giảm số lượng "jailbreaks" có thể mở khóa các khả năng mạnh mẽ hơn của mô hình. Động thái này đánh dấu một sự thay đổi trong cách tiếp cận của Anthropic để giải quyết các lo ngại về bảo mật của chính quyền, những lo ngại mà công ty trước đây đã phản đối. Thỏa thuận này được thương lượng bởi Lutnick và giám đốc an ninh mạng quốc gia Sean Cairncross, và được coi là một bước phát triển quan trọng trong nỗ lực của công ty để đưa các mô hình AI của mình trở lại trực tuyến.

6

ScarfBench: Benchmarking AI Agents cho việc di chuyển Enterprise Java Framework

ScarfBench: Benchmarking AI Agents for Enterprise Java Framework Migration

Hugging Face Blog [Đọc bài viết →](#)

Việc hiện đại hóa các ứng dụng doanh nghiệp là một quá trình phức tạp và tốn kém, với các đội di chuyển ứng dụng qua các framework để cải thiện khả năng bảo trì và truy cập vào các khả năng hiện đại. Những tiến bộ gần đây trong các tác nhân mã hóa đã tạo ra sự phấn khích xung quanh việc hiện đại hóa được hỗ trợ bởi AI, nhưng một câu hỏi quan trọng vẫn còn: liệu các tác nhân AI có thể hiện đại hóa đáng tin cậy các ứng dụng doanh nghiệp thực tế hay không? Để giải quyết khoảng trống này, các nhà nghiên cứu đã giới thiệu ScarfBench, một điểm chuẩn mở để đánh giá các tác nhân AI trên các nhiệm vụ di chuyển framework trong Enterprise Java. ScarfBench tập trung vào việc di chuyển qua ba hệ sinh thái Java chính: Spring, Jakarta EE và Quarkus. Không giống như các điểm chuẩn truyền thống, ScarfBench đánh giá liệu các ứng dụng đã di chuyển có thực sự được xây dựng, triển khai và bảo tồn hành vi hay không, cung cấp một biện pháp thực tế hơn về chất lượng hiện đại hóa. Các nhà nghiên cứu đã đánh giá một số tác nhân mã hóa tiên tiến nhất trên ScarfBench và phát hiện ra rằng việc di chuyển framework vẫn còn khó khăn, với tỷ lệ thành công khác nhau đáng kể trên các cặp framework và các lần di chuyển toàn ứng dụng. Sự thành công khi biên dịch nhất quán vượt qua sự thành công khi triển khai, điều này lại vượt qua sự thành công về hành vi, nhấn mạnh tầm quan trọng của việc đánh giá không chỉ việc chuyển đổi mã mà còn cả các quá trình xây dựng và triển khai. Nghiên cứu cũng gợi ý rằng việc tự đánh giá của tác nhân không nên được coi là một tín hiệu đáng tin cậy về việc hoàn thành di chuyển, và rằng di chuyển là một quá trình giải quyết sự phụ thuộc lặp lại chứ không phải là một chuyển đổi nguồn-thành-nguồn đơn giản.

7

Function Calling từ Angular — Structured JSON từ LLMs (2026)

Function Calling from Angular — Structured JSON from LLMs (2026)

Dev.to AI [Đọc bài viết →](#)

Một hướng dẫn mới đã được xuất bản trên DEV Community, với tiêu đề "Function Calling from Angular — Structured JSON từ LLMs (2026)". Hướng dẫn này trình diễn cách sử dụng Angular để gọi hàm đối với

backend Spring AI, cụ thể là sử dụng phương pháp @Tool. Hướng dẫn cũng bao gồm việc sử dụng khối @if trong mẫu để hiển thị thẻ giá khi trợ lý trả về JSON. Ngoài ra, hướng dẫn cũng đề cập đến các chủ đề liên quan như xác thực biểu mẫu phản ứng cho xác thực đầu vào trước khi gửi lời nhắc. Mã code đầy đủ cho hướng dẫn có sẵn trên GitHub. Hướng dẫn này là một phần của loạt hướng dẫn AI Developer Tutorials và được thiết kế để giúp các developer nhanh chóng trả lời các câu hỏi thường gặp hoặc lưu trữ các đoạn code để tái sử dụng.

8

Morgan Stanley giảm một nửa công việc đối chiếu rủi ro nhất nhờ giảm tính tự chủ của các agent

Morgan Stanley cut its riskiest reconciliation job in half — by making its agents less autonomous

VentureBeat [Đọc bài viết →](#)

Morgan Stanley đã giảm thành công thời gian dành cho việc đối chiếu lợi nhuận và lỗ (P&L) xuống một nửa, từ sáu giờ xuống hai đến ba giờ, bằng cách triển khai một hệ thống sản xuất nội bộ có tên là FIXR. Hệ thống này, bao gồm nhiều tác nhân làm việc cùng nhau, phân tích "lỗi" trong dữ liệu và đề xuất giải pháp dựa trên các quy tắc đã học. Tuy nhiên, không giống như nhiều triển khai AI, FIXR không dựa vào tính tự chủ, mà dựa trên vòng phản hồi giữa con người và tác nhân, nơi con người xem xét, phê duyệt hoặc sửa đổi mọi khuyến nghị do hệ thống đưa ra. Cách tiếp cận này cho phép hệ thống học hỏi từ người kiểm soát và cải thiện theo thời gian, đồng thời vẫn duy trì trách nhiệm giải trình của con người. Hệ thống được thiết kế để xác định và lặp lại, với trọng tâm là quy trình trước và khả năng mở rộng. Bằng cách thiết lập quy trình trước và sau đó thêm AI, Morgan Stanley đã có thể xác định các lĩnh vực mà tự động hóa sẽ mang lại lợi ích nhất và biến cơ hội thành hiện thực. Kinh nghiệm của công ty phản ánh các mẫu được thấy trong các triển khai AI doanh nghiệp, nơi các phương pháp quy trình trước và mua-và-kết hợp có thể bền vững hơn so với việc theo đuổi các model tùy chỉnh.

TIPS & TRICKS CHO DEV

Kết nối File System

Vấn đề: Cần truy cập file system trong Claude Desktop.

Cách làm: Sử dụng MCP để kết nối với file system, cho phép đọc và ghi file. Ví dụ:

Claude, tạo file mới trong thư mục hiện tại.

Đánh giá: Hiệu quả cao, nên dùng khi cần tương tác với file system.

Tích hợp Git

Vấn đề: Cần tích hợp Git vào Claude Code.

Cách làm: Sử dụng MCP để kết nối với Git, cho phép thực hiện lệnh Git. Ví dụ:

```
git clone https://github.com/user/repo
```

Đánh giá: Nên dùng khi cần quản lý mã nguồn.

Truy cập Database

Vấn đề: Cần truy cập database trong Cursor.

Cách làm: Sử dụng MCP để kết nối với database, cho phép thực hiện lệnh SQL. Ví

dụ:

```
SELECT * FROM table
```

.

Đánh giá: Hiệu quả cao, nên dùng khi cần tương tác với database.

BÀI HỌC AI HÔM NAY CHO DEV

1. Tối ưu chi phí & hiệu năng LLM

2. Trong quá trình phát triển ứng dụng AI, việc tối ưu hóa chi phí và hiệu năng của mô hình ngôn ngữ lớn (LLM) là rất quan trọng. Điều này giúp giảm thiểu chi phí vận hành và cải thiện hiệu suất của ứng dụng. Dev cần biết cách tối ưu hóa LLM để đảm bảo ứng dụng của họ hoạt động hiệu quả.

3. Ví dụ, việc sử dụng kỹ thuật fine-tuning và LoRA (Low-Rank Adaptation) có thể giúp giảm kích thước mô hình và cải thiện hiệu suất. Code snippet minh họa:

```
model = LLM.from_pretrained('base_model'); model.fine_tune(...)
```

4. Tip hoặc bước tiếp theo: Để tối ưu hóa LLM, dev nên thử nghiệm với các kỹ thuật khác nhau như quantization, pruning và knowledge distillation để tìm ra phương pháp phù hợp nhất cho ứng dụng của mình.

Luôn đi đầu trong thế giới AI! · Stay ahead in AI!

Nguồn: Google News · Groq AI