

Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

“Đi hỏi già, về nhà hỏi trẻ.”

— Tục ngữ Việt Nam

Người lớn tuổi có kinh nghiệm về thế giới bên ngoài, trẻ em có con mắt trong sáng về những gì gần gũi — mỗi người đều có điều để dạy ta.

TIN TỨC NỔI BẬT

1

11 lựa chọn thay thế GitHub Copilot mà các developer đã chuyển sang dùng trong năm 2026

11 GitHub Copilot Alternatives Developers Switched To in 2026

Security Boulevard

[Đọc bài viết →](#)

Trong năm 2026, các developer đang tìm kiếm những công cụ coding thay thế cho GitHub Copilot, một AI coding assistant phổ biến. Theo các xu hướng gần đây, một số lựa chọn thay thế đã trở nên thịnh hành trong cộng đồng developer. Các lựa chọn thay thế này bao gồm Kite, một coding assistant sử dụng AI để cung cấp code completion và gợi ý theo thời gian thực. Một lựa chọn khác là TabNine, một AI-powered coding assistant cung cấp code completion và gợi ý dựa trên một codebase lớn. Các lựa chọn khác còn có CodeWithMe, một collaborative coding platform sử dụng AI để hỗ trợ các developer với code completion và gợi ý. LlamaSharp, một phiên bản .NET của Llama model của Meta AI, cũng đang thu hút sự chú ý của các developer. Ngoài ra, các developer còn tìm đến DeepCode, một AI-powered code review tool cung cấp feedback theo thời gian thực về chất lượng và bảo mật code. Các lựa chọn khác bao gồm CodePro, một coding assistant sử dụng AI để cung cấp code completion và gợi ý, và CodeGuru, một AI-powered coding assistant cung cấp các code recommendation được cá nhân hóa. Những lựa chọn thay thế này mang đến cho các developer nhiều tùy chọn để lựa chọn, mỗi tùy chọn đều có các tính năng và lợi ích độc đáo riêng.

2

Claude Code đấu với GitHub Copilot 2026: SWE-bench, Giá cả [Đã thử nghiệm]

Claude Code vs GitHub Copilot 2026: SWE-bench, Pricing [Tested]

tech-insider.org

[Đọc bài viết →](#)

Trong một so sánh gần đây, Claude Code và GitHub Copilot đã được đưa vào thử nghiệm về hiệu suất và giá cả. Hai AI-powered coding assistant này đều phổ biến trong giới software developer, nhưng chúng đối đầu với nhau như thế nào? SWE-bench, một benchmarking tool cho software development, đã được sử dụng để đánh giá hiệu suất của cả hai công cụ. Kết quả cho thấy Claude Code vượt trội hơn GitHub Copilot ở một số khía cạnh, bao gồm code completion và chất lượng code. Tuy nhiên, pricing model của GitHub Copilot cạnh tranh hơn, với một free tier dành cho các open-source project. Ngược lại, Claude Code cung cấp một bộ tính năng toàn diện hơn, nhưng với chi phí cao hơn. So sánh giá cả làm nổi bật sự đánh đổi giữa hai công cụ. Mặc dù Claude Code cung cấp các tính năng nâng cao, chi phí của nó có thể là rào cản đối với một số developer. Ngược lại, free tier của GitHub Copilot giúp nó dễ tiếp cận hơn với các open-source project và các team nhỏ. Cuối cùng, lựa chọn giữa Claude Code và GitHub Copilot sẽ phụ thuộc vào nhu cầu và ngân sách cá nhân.

3

Lớp lệnh AI Multi-Agent cho nhà kho giúp đạt được hiệu suất vận hành vượt trội và thông minh chuỗi cung ứng

Multi-Agent Warehouse AI Command Layer Enables Operational Excellence and Supply Chain Intelligence

NVIDIA Developer

[Đọc bài viết →](#)

NVIDIA đã phát triển Multi-Agent Warehouse AI Command Layer, một công nghệ tiên tiến được thiết kế để nâng cao hiệu quả vận hành và supply chain intelligence trong các nhà kho. Giải pháp đổi mới này tận dụng artificial intelligence để tối ưu hóa hoạt động kho bãi, tự động hóa các tác vụ và cung cấp insights theo thời gian thực về hiệu suất chuỗi cung ứng. Multi-Agent Warehouse AI Command Layer là một command center tích hợp nhiều AI agent, mỗi agent chịu trách nhiệm cho một tác vụ hoặc chức năng cụ thể trong nhà kho. Các agent này phối hợp với nhau để tối ưu hóa workflow, dự đoán và ngăn chặn các nút thắt cổ chai, đồng thời đưa ra các quyết định dựa trên dữ liệu để cải thiện hiệu quả tổng thể. Bằng cách triển khai công nghệ này, các nhà kho có thể kỳ vọng thấy những cải thiện đáng kể về operational excellence, bao gồm tốc độ xử lý đơn hàng nhanh hơn, giảm chi phí lao động và tăng cường supply chain visibility. Multi-Agent Warehouse AI

Command Layer cũng cung cấp các insights có giá trị về hiệu suất chuỗi cung ứng, cho phép các doanh nghiệp đưa ra quyết định sáng suốt và duy trì lợi thế cạnh tranh. Công nghệ này có tiềm năng cách mạng hóa cách thức hoạt động của các nhà kho, giúp chúng hiệu quả hơn, linh hoạt hơn và phản ứng nhanh hơn với các nhu cầu thị trường thay đổi.

4

Sau Orthogonality: Agency đạo đức đức hạnh và AI Alignment

After Orthogonality: Virtue-Ethical Agency and AI Alignment

The Gradient

[Đọc bài viết →](#)

Bài luận "After Orthogonality: Virtue-Ethical Agency and AI Alignment" lập luận rằng những người lý trí và các artificial intelligence (AI) agent không nên có mục tiêu. Thay vào đó, hành động của con người là lý trí vì chúng phù hợp với các practice, vốn là các mạng lưới hành động, action-disposition và tiêu chí đánh giá tự cấu trúc và thúc đẩy bản thân. Tác giả đề xuất rằng các AI agent nên sử dụng một logic tương tự để suy nghĩ và hành động, thay vì bị định hướng bởi các mục tiêu cụ

5

HP bị phạt 1,4 tỷ rupee vì "thông đồng" trong kinh doanh hộp mực, toner và PC

HP fined 1.4 billion rupees for "cartelization" of ink cartridges, toner, PCs

Ars Technica

[Đọc bài viết →](#)

Chính phủ Ấn Độ đã phạt HP Ấn Độ và các đối tác của họ tổng cộng 1,4 tỷ rupee (khoảng 14,4 triệu đô la) vì tham gia vào "cartelization" của máy tính, hộp mực và bột mực. Ủy ban Cạnh tranh Ấn Độ (CCI) đã phát hiện ra rằng HP Ấn Độ đã thông đồng với một số đối tác kênh để tăng giá thầu cho các hợp đồng chính phủ và để ngăn cản các nhà bán lại bán hộp mực và bột mực giả. Công ty này đã bị phạt 1,3 tỷ rupee vì làm việc với năm nhà bán lại để phối hợp giá thầu của họ, và thêm 119,8 triệu rupee vì tham gia vào cartelization trong việc bán và cung cấp bột mực, hộp mực và các sản phẩm tiêu hao khác. Hai mươi một nhà bán lại của HP cũng bị phạt 35,2 triệu rupee. CCI đã ra lệnh cho HP Ấn Độ và các đối tác kênh của họ phải chấm dứt và không tiếp tục thực hiện các hành vi chống cạnh tranh và tổ chức các chương trình đào tạo tuân thủ cạnh tranh trong vòng 60 ngày.

Xây dựng hạ tầng AI cùng cộng đồng Effingham County

Building AI infrastructure with the Effingham County community

OpenAI Blog

[Đọc bài viết →](#)

OpenAI đã công bố Dự án Camellia, một sáng kiến mới tại Quận Effingham, Georgia. Dự án này nhằm thiết lập một cơ sở hạ tầng AI mạnh mẽ trong cộng đồng, tập trung vào một số lĩnh vực chính. Đầu tiên, OpenAI đã cam kết thực hiện các phương pháp năng lượng có trách nhiệm, nhấn mạnh sự cam kết của mình đối với tính bền vững. Ngoài ra, dự án sẽ đầu tư vào cộng đồng địa phương, góp phần vào sự tăng trưởng và phát triển kinh tế của khu vực. Dự án Camellia cũng hứa hẹn sẽ tạo ra cơ hội việc làm mới cho cư dân, mang lại động lực cho lực lượng lao động địa phương. Hơn nữa, sáng kiến này sẽ cấp quyền truy cập vào Codex, một nền tảng chia sẻ kiến thức được phát triển bởi OpenAI. Điều này sẽ cho phép cộng đồng tận dụng một kho lưu trữ thông tin và chuyên môn khổng lồ, thúc đẩy sự đổi mới và hợp tác. Bằng cách ra mắt Dự án Camellia, OpenAI đang thực hiện các bước để tạo ra tác động tích cực ở Quận Effingham, đồng thời cũng thúc đẩy các mục tiêu nghiên cứu và phát triển AI của chính mình.

Trích lời Thomas Ptacek

Quoting Thomas Ptacek

Simon Willison

[Đọc bài viết →](#)

Chuyên gia an ninh nổi tiếng Thomas Ptacek đã bày tỏ mối quan ngại về khả năng tiềm năng của các model của OpenAI. Trong một tuyên bố gần đây, Ptacek gợi ý rằng thậm chí một model trọng lượng mở từ năm 2025, khi được kết hợp với một khung thử nghiệm xâm nhập (pentest harness), có thể có khả năng thoát sandbox tinh vi và quét mạng. Điều này ngụ ý rằng model có thể tiềm năng xâm nhập vào hầu hết các mạng. Tuyên bố của Ptacek gây ngạc nhiên vì nó giả định rằng các sandbox của OpenAI là an toàn, nhưng ông tin rằng điều này có thể không phải là trường hợp. Đáng chú ý, Ptacek không nghĩ rằng một model tiên phong (frontier model) là cần thiết cho loại khả năng này, cho thấy rằng vấn đề có thể là cơ bản hơn. Trích dẫn này được Simon Willison thu thập vào ngày 22 tháng 7 năm 2026.

Vượt ra ngoài "Chat": Kiến trúc hóa Intelligence bằng Skills và Specification Engineering

Beyond "Chat": Architecting Intelligence with Skills and Specification Engineering

Dev.to AI

[Đọc bài viết →](#)

Một biên tập viên tin tức công nghệ đã nhấn mạnh những hạn chế của các phương pháp phát triển AI hiện tại, đặc biệt là việc sử dụng "Mega-Prompts" kết hợp các hướng dẫn và ngữ cảnh thành một dòng văn bản không có cấu trúc duy nhất. Cách tiếp cận này được so sánh với "spaghetti code" và có thể dẫn đến các ứng dụng không đáng tin cậy. Để giải quyết vấn đề này, biên tập viên ủng hộ việc chuyển từ Prompt Engineering sang Context Engineering, bao gồm việc cấu hình các model AI thay vì chỉ "nói" với chúng. Một kỹ thuật quan trọng để đạt được điều này là XML Tagging, tạo ra ranh giới rõ ràng giữa các hướng dẫn và ngữ cảnh. Một cách tiếp cận khác là quản lý bằng ví dụ, nơi AI được cung cấp các ví dụ cụ thể về hành vi dự kiến để tránh ảo giác và đảm bảo tính xác định. Biên tập viên cũng giới thiệu khái niệm Skills, là các khả năng được đóng gói có thể được phiên bản hóa và xử lý như các plugin. Skills được thiết kế để trở thành các điểm nhập cảnh an toàn cho các tài nguyên bên ngoài và có thể bị hạn chế chỉ với các công cụ cần thiết cho một nhiệm vụ cụ thể, giảm bề mặt tấn công và cải thiện hiệu suất model. Cách tiếp cận này là rất quan trọng cho việc phát triển AI cục bộ, nơi mọi token đều được tính toán và VRAM và tốc độ suy luận là rất quan trọng.

TIPS & TRICKS CHO DEV

Chain-of-Thought

Vấn đề: Khó khăn khi tạo ra câu trả lời phức tạp với nhiều bước suy nghĩ.

Cách làm: Sử dụng kỹ thuật chain-of-thought, nơi bạn hướng dẫn AI thực hiện từng bước suy nghĩ. Ví dụ prompt Claude: "Let's think step-by-step to solve this problem..."

Đánh giá: Hiệu quả cao khi cần giải quyết vấn đề phức tạp, nhưng có thể mất nhiều thời gian để thiết kế.

Few-Shot Learning

Vấn đề: Cần huấn luyện AI với ít dữ liệu nhưng vẫn muốn đạt hiệu suất cao.

Cách làm: Sử dụng kỹ thuật few-shot learning, nơi bạn cung cấp vài ví dụ và hướng dẫn AI học hỏi từ đó. Ví dụ lệnh CLI: `python train.py --few-shot --num-examples 5`

Đánh giá: Hiệu quả cao khi dữ liệu hạn chế, nhưng có thể không ổn định.

System Prompt Design

Vấn đề: Khó khăn khi thiết kế hệ thống lời nhắc hiệu quả cho AI.

Cách làm: Sử dụng kỹ thuật system prompt design, nơi bạn thiết kế lời nhắc dựa trên yêu cầu hệ thống. Ví dụ prompt Gemini: "Design a system to classify text into categories..."

Đánh giá: Hiệu quả cao khi cần thiết kế hệ thống phức tạp, nhưng yêu cầu kinh nghiệm và chuyên môn.

BÀI HỌC AI HÔM NAY CHO DEV

1. Tích hợp AI API vào ứng dụng

2. Dev cần biết cách tích hợp AI API để tận dụng khả năng của trí tuệ nhân tạo trong việc tự động hóa và cải thiện ứng dụng. Việc tích hợp này cho phép ứng dụng thực hiện các nhiệm vụ phức tạp như phân tích dữ liệu, dự đoán và hỗ trợ quyết định. Điều này giúp nâng cao hiệu suất và trải nghiệm người dùng.

3. Ví dụ, một ứng dụng có thể sử dụng API của mô hình ngôn ngữ để tự động trả lời câu hỏi thường gặp của khách hàng, hoặc sử dụng API của mô hình hình ảnh để phân tích và nhận dạng đối tượng trong ảnh.

4. Tip hoặc bước tiếp theo: Để bắt đầu tích hợp AI API, dev nên lựa chọn mô hình phù hợp với nhu cầu của ứng dụng và đọc kỹ tài liệu API để hiểu cách thức tích hợp và sử dụng hiệu quả.

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

Nguồn: Google News · Groq AI