BIS - MT · 13/05/2026

🤖 Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

✨ *"Talk is cheap. Show me the code."*

↪ Nói thì rẻ. Hãy cho tôi xem code.

— Linus Torvalds

💡 *Trong kỹ thuật phần mềm, hành động và kết quả thực tế quan trọng hơn ý tưởng hay lời hứa — hãy chứng minh qua sản phẩm cụ thể.*

📰 TIN TỨC NỔI BẬT

1

So sánh Claude Code và 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 →

> Một so sánh gần đây đã được thực hiện giữa Claude Code và GitHub Copilot, hai công cụ mã hóa được hỗ trợ bởi AI phổ biến. Công cụ SWE-bench, một công cụ chuẩn để kỹ thuật phần mềm, đã được sử dụng để đánh giá hiệu suất của cả hai công cụ. Kết quả cho thấy Claude Code vượt trội so với GitHub Copilot ở nhiều lĩnh vực, bao gồm chất lượng mã và hiệu quả. Về giá cả, GitHub Copilot cung cấp một tầng miễn phí với các tính năng hạn chế, cũng như một đăng ký trả phí có giá 10 đô la mỗi tháng. Claude Code, mặt khác, cung cấp một thử nghiệm miễn phí, nhưng mô hình giá của nó không được nêu rõ trong so sánh. So sánh cũng nhấn mạnh sự khác biệt trong trải nghiệm người dùng và giao diện của hai công cụ. Mặc dù cả hai công cụ đều nhằm hỗ trợ các nhà phát triển với việc hoàn thành mã và đề xuất, chúng có những cách tiếp cận khác nhau để đạt được mục tiêu này. Kết quả của so sánh gợi ý rằng Claude Code có thể là một lựa chọn hiệu quả hơn cho các nhà phát triển đang tìm kiếm một trải nghiệm mã hóa chất lượng cao.

2

Cách chạy các AI model cục bộ trên RTX 5060 Ti của bạn — Hướng dẫn từng bước

🇬🇧 *How to Run Local AI Models on Your RTX 5060 Ti — Step-by-Step Guide*

📰 TechnoSports Media Group 🔗 Đọc bài viết →

> Hướng dẫn từng bước đã được xuất bản để giúp người dùng chạy các mô hình AI cục bộ trên card đồ họa NVIDIA GeForce RTX 5060 Ti của họ. Hướng dẫn cung cấp hướng dẫn toàn diện cho những cá nhân muốn tận dụng khả năng AI của RTX 5060 Ti. Để bắt đầu, người dùng cần cài đặt phần mềm cần thiết, bao gồm CUDA, cuDNN và TensorRT. Những công cụ này cho phép RTX 5060 Ti chạy các mô hình AI hiệu quả. Tiếp theo, người dùng phải tải xuống và cài đặt Deep Learning SDK, cung cấp một framework cho việc phát triển và triển khai các mô hình AI. Một khi phần mềm đã được cài đặt, người dùng có thể bắt đầu xây dựng và đào tạo các mô hình AI của họ bằng cách sử dụng các framework phổ biến như TensorFlow hoặc PyTorch. Hướng dẫn sau đó hướng dẫn người dùng qua quá trình triển khai các mô hình của họ trên RTX 5060 Ti, bao gồm cấu hình mô hình và tối ưu hóa hiệu suất. Bằng cách làm theo hướng dẫn này, người dùng có thể mở khóa toàn bộ tiềm năng của RTX 5060 Ti và chạy các mô hình AI cục bộ một cách dễ dàng. Hướng dẫn được thiết kế cho người dùng có kiến thức kỹ thuật nhất định và cung cấp một cách tiếp cận chi tiết, từng bước để bắt đầu phát triển AI trên RTX 5060 Ti.

3 OpenAI ra mắt Codex Chrome Extension cho các tác vụ web development chạy nền

🇬🇧 *OpenAI Launches Codex Chrome Extension for Background Web Development Tasks*

📰 Technobezz 🔗 Đọc bài viết →

> OpenAI đã ra mắt một tiện ích mở rộng Chrome mới có tên là Codex, được thiết kế để hỗ trợ các nhiệm vụ phát triển web ở chế độ nền. Tiện ích mở rộng này sử dụng model AI Codex của OpenAI để tự động hóa các nhiệm vụ lập trình lặp đi lặp lại, cho phép các developer tập trung vào các khía cạnh phức tạp và sáng tạo hơn của công việc. Với tiện ích mở rộng Codex của Chrome, người dùng có thể thực hiện các nhiệm vụ như hoàn thành mã, gỡ lỗi và tái cấu trúc, tất cả đều ở chế độ nền. Điều này cho phép các developer làm việc hiệu quả hơn và giải phóng thời gian cho các nhiệm vụ cấp cao hơn. Tiện ích mở rộng này tương thích với nhiều ngôn ngữ lập trình, bao gồm Python, JavaScript và HTML/CSS. Tiện ích mở rộng Codex của Chrome có sẵn để tải xuống trên Chrome Web Store, và người dùng có thể truy cập các tính năng của nó bằng cách nhấp vào biểu tượng của tiện ích mở rộng trong thanh công cụ trình duyệt của họ. Model AI Codex của OpenAI được đào tạo trên một tập dữ liệu mã lớn và có khả năng hiểu các sắc thái của các ngôn ngữ lập trình khác nhau. Bằng cách tận

dụng công nghệ này, tiện ích mở rộng Codex của Chrome nhằm mục đích tối ưu hóa các quy trình phát triển web và tăng cường năng suất.

4

AGI không phải là Multimodal

🇬🇧 *AGI Is Not Multimodal*

📰 The Gradient 🔗 Đọc bài viết →

Những tiến bộ gần đây trong các mô hình AI tạo sinh đã khiến một số người tin rằng Trí tuệ Tổng quát Nhân tạo (AGI) sắp trở thành hiện thực. Tuy nhiên, sự lạc quan này có thể là không phù hợp. Sự thành công của những mô hình này chủ yếu là do khả năng mở rộng trên phần cứng hiện có, chứ không phải là một cách tiếp cận có suy nghĩ để giải quyết vấn đề về trí tuệ. Cách tiếp cận đa phương thức, kết hợp nhiều phương thức để tạo ra một AI tổng quát, không thể dẫn đến AGI ở mức độ con người. Chiến lược này tập trung vào việc xử lý từng phương thức riêng biệt, thay vì coi sự hiện diện và tương tác với môi trường là yếu tố chính. Một AGI thực sự phải là tổng quát trên tất cả các lĩnh vực, bao gồm cả thực tại vật lý, và có khả năng giải quyết các vấn đề xuất phát từ thế giới vật lý. Các Mô hình Ngôn ngữ Lớn (LLM) hiện tại có thể có một sự hiểu biết bề mặt về thực tại, chứ không phải là một sự hiểu biết sâu sắc, và khả năng phản ánh một sự hiểu biết giống con người về thế giới không nhất thiết là dấu hiệu của trí tuệ thực sự.

5

Sam Altman tiết lộ: Elon Musk từng có ý tưởng "rợn tóc gáy" về việc chuyển giao OpenAI cho con cái

🇬🇧 *Elon Musk Had 'Hair-Raising' Idea of Passing OpenAI Onto His Kids, Sam Altman Says*

📰 Wired 🔗 Đọc bài viết →

Trong vụ xét xử Musk v. Altman đang diễn ra, CEO OpenAI Sam Altman đã bước lên bục chứng nhân để bảo vệ danh tiếng của mình trước những cáo buộc về hành vi lừa đảo. Trong lời khai của mình, Altman đã tiết lộ một khoảnh khắc "gây rợn" khi Elon Musk đề xuất rằng quyền kiểm soát OpenAI nên được chuyển cho con cái của ông nếu ông qua đời. Altman cho biết rằng ông và Tổng thống OpenAI Greg Brockman không cảm thấy thoải mái với đề xuất này. Vụ xét xử này tập trung vào vụ kiện của Musk, cáo buộc Altman lạm dụng 38 triệu đô la được quyên góp cho OpenAI, biến tổ chức từ thiện này thành một doanh nghiệp có lợi nhuận trị giá hơn 850 tỷ đô la. Tuy

nhiên, Altman và Brockman đã làm chứng rằng họ không nhớ Musk đã gắn bất kỳ điều kiện đặc biệt nào với các khoản quyên góp của mình. Ngoài ra, dường như Musk có thể đã nộp đơn kiện quá muộn, vì thời hiệu đã hết hạn. Altman tự miêu tả mình là một doanh nhân và nhà đầu tư quan tâm đến sức mạnh của AI, trong khi luật sư của Musk, Steven Molo, đã đặt câu hỏi về tính đáng tin cậy của Altman, dẫn chứng các cáo buộc từ các đồng nghiệp và đối tác kinh doanh cũ.

6

Bài học về UX, security và scale khi xây dựng một Slack agent cấp doanh nghiệp

🇬🇧 *Lessons on UX, security, and scale when building an enterprise-grade Slack agent*

📰 Sourcegraph Blog 🔗 Đọc bài viết →

Một đội đã phát triển một đại lý Slack cấp doanh nghiệp, được thiết kế đặc biệt cho các công ty lớn. Đại lý này, được gọi là Deep Search, được xây dựng để cung cấp những thông tin và bài học quý giá về trải nghiệm người dùng, bảo mật và khả năng mở rộng. Về mặt trải nghiệm người dùng, đội ngũ đã tập trung vào việc tạo ra một giao diện trực quan đáp ứng nhu cầu của các doanh nghiệp lớn. Thiết kế của đại lý này nhấn mạnh sự dễ sử dụng, khả năng tiếp cận và tích hợp liền mạch với các quy trình làm việc hiện có. Từ góc độ bảo mật, đội ngũ đã ưu tiên các biện pháp bảo vệ mạnh mẽ để bảo vệ dữ liệu công ty nhạy cảm. Điều này bao gồm việc triển khai các tính năng bảo mật cấp doanh nghiệp để ngăn chặn truy cập không được phép và vi phạm dữ liệu. Ngoài ra, đội ngũ đã sử dụng kỹ thuật hạn chế tốc độ dựa trên Redis, một kỹ thuật được sử dụng để điều chỉnh số lượng yêu cầu gửi đến đại lý. Điều này giúp ngăn chặn lạm dụng và đảm bảo đại lý vẫn phản hồi và hiệu quả, ngay cả khi sử dụng nặng. Những bài học rút ra từ việc xây dựng Deep Search có thể trở thành một nguồn tài nguyên quý giá cho các nhà phát triển và công ty muốn tạo ra các đại lý Slack của riêng họ cho các doanh nghiệp lớn, sử dụng các công nghệ như AI, API, LLM, model, token, developer, framework, v.v.

7

Các đội ngũ tài chính sử dụng Codex như thế nào

🇬🇧 *How finance teams use Codex*

📰 OpenAI Blog 🔗 Đọc bài viết →

Các đội tài chính có thể tận dụng Codex để tự động hóa và đơn giản hóa các nhiệm vụ và quy trình tài chính khác nhau. Một ứng dụng

quan trọng của Codex là xây dựng Báo cáo Cân đối Quản lý (MBRs), cung cấp cái nhìn tổng thể về hiệu suất tài chính của một tổ chức. Ngoài ra, các đội tài chính có thể sử dụng Codex để tạo các gói báo cáo, cho phép họ trình bày dữ liệu tài chính một cách rõ ràng và có tổ chức. Codex cũng cho phép các đội tài chính xây dựng cầu chênh lệch, giúp xác định và phân tích sự khác biệt giữa dữ liệu tài chính thực tế và dự báo. Hơn nữa, nền tảng này cho phép kiểm tra model, đảm bảo rằng các model tài chính là chính xác và đáng tin cậy. Hơn nữa, các đội tài chính có thể sử dụng Codex để tạo các kịch bản lập kế hoạch, cho phép họ mô phỏng các kết quả tài chính khác nhau và đưa ra quyết định thông minh. Bằng cách sử dụng Codex, các đội tài chính có thể tự động hóa và đơn giản hóa các nhiệm vụ tài chính khác nhau, giải phóng nguồn lực cho các hoạt động chiến lược và có giá trị cao hơn. Nền tảng này cung cấp một công cụ mạnh mẽ cho các đội tài chính để quản lý và phân tích dữ liệu tài chính, cuối cùng thúc đẩy việc ra quyết định và kết quả kinh doanh tốt hơn.

8

Bitwarden CLI bị xâm nhập

🇬🇧 *Bitwarden CLI compromised*

📰 Changelog 🔗 Đọc bài viết →

Không có đề cập nào về việc Bitwarden CLI bị xâm phạm trong nội dung được cung cấp. Văn bản dường như là một lời chứng thực về Changelog Newsletter, ca ngợi nội dung, định dạng và người chủ trì, Jerod. Người viết thể hiện sự đánh giá cao của họ đối với bản tin, tuyên bố nó là một nguồn thông tin quý giá và một lợi thế cạnh tranh.

⚡ TIPS & TRICKS CHO DEV

⚡ Sử dụng LangGraph

Vấn đề: Xử lý task phức tạp cần phối hợp nhiều AI agents.

Cách làm: Sử dụng LangGraph để phối hợp các agents, ví dụ: "Tôi cần viết một đoạn văn bằng tiếng Anh về chủ đề AI".

Đánh giá: Hiệu quả khi cần tích hợp nhiều agents, nhưng cần cấu hình phù hợp.

⚡ Tích hợp CrewAI

Vấn đề: Cần tự động hóa quy trình xử lý task.

Cách làm: Tích hợp CrewAI vào hệ thống, sử dụng lệnh "crewai automate" để tự động hóa quy trình.

Đánh giá: Hiệu quả khi cần tự động hóa quy trình, nhưng cần giám sát thường xuyên.

⚡ Áp dụng AutoGen

Vấn đề: Cần□□ dữ liệu nhanh chóng và chính xác.
Cách làm: Sử dụng AutoGen với lệnh "autogen generate" để tạo dữ liệu.
Đánh giá: Hiệu quả khi cần dữ liệu nhanh chóng, nhưng cần kiểm tra chất lượng dữ liệu.

📚 BÀI HỌC AI HÔM NAY CHO DEV

1. **Tối ưu chi phí & hiệu năng LLM**
2. Trong phát triển ứng dụng AI, việc tối ưu chi phí và hiệu năng của Large Language Model (LLM) là rất quan trọng để đảm bảo hiệu suất và tiết kiệm tài nguyên. Dev cần biết cách tối ưu hóa LLM để cải thiện hiệu suất và giảm chi phí.
3. Ví dụ, sử dụng kỹ thuật fine-tuning và LoRA (Low-Rank Adaptation) có thể giúp giảm thiểu kích thước mô hình và cải thiện hiệu suất. Ví dụ code: `model = transformers.AutoModelForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=8)`.
4. 💡 Tip hoặc bước tiếp theo: Sử dụng thư viện như Hugging Face Transformers để thực hiện fine-tuning và LoRA cho LLM, và đánh giá hiệu suất của mô hình trên các tập dữ liệu cụ thể.