BIS - MT · 16/06/2026

Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

“It's not about how hard you hit. It's about how hard you can get hit and keep moving forward.”

↪ Không phải về việc bạn đánh mạnh đến đâu. Mà là bạn có thể chịu đựng đòn nặng thế nào và vẫn tiếp tục tiến về phía trước.

— Rocky Balboa

Sức bền và khả năng phục hồi quan trọng hơn sức tấn công — người thật sự mạnh là người không bỏ cuộc dù bị đánh gục nhiều lần.

TIN TỨC NỔI BẬT

1

Claude Code đấu GitHub Copilot 2026: SWE-bench, Giá cả [Kiểm chứng]

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 khả năng kỹ thuật phần mềm và giá cả. SWE-bench, một công cụ benchmark cho kỹ thuật phần mềm, đã được sử dụng để đánh giá hiệu suất của cả hai trợ lý lập trình được hỗ trợ bởi AI. Kết quả cho thấy Claude Code vượt trội hơn GitHub Copilot ở một số lĩnh vực, bao gồm code completion, debugging và code review. Claude Code thể hiện tỷ lệ chính xác cao hơn và thời gian phản hồi nhanh hơn, biến nó thành một lựa chọn hiệu quả hơn cho các developer. Về giá cả, GitHub Copilot cung cấp một free tier với các tính năng giới hạn, cũng như một gói đăng ký trả phí bắt đầu từ 10 USD mỗi tháng. Claude Code, mặt khác, cung cấp bản dùng thử miễn phí và một gói trả phí với cấu trúc giá cạnh tranh hơn. So sánh này làm nổi bật điểm mạnh và điểm yếu của từng công cụ, cung cấp cho các developer những thông tin chi tiết có giá trị để đưa ra quyết định sáng suốt về trợ lý lập trình AI nào phù hợp nhất với nhu cầu của họ.

2

Ai cũng muốn bắt kịp xu hướng vibe coding — và Google cũng không ngoại lệ với Stitch, dự án tiếp nối Jules

Everyone's looking to get in on vibe coding — and Google is no different with Stitch, its follow-up to Jules

VentureBeat Đọc bài viết →

> Google đang gia nhập không gian vibe coding với Stitch, dự án tiếp nối của Jules. Vibe coding đề cập đến việc viết code mang tính biểu cảm và sáng tạo hơn, thường lấy cảm hứng từ âm nhạc và các loại hình nghệ thuật khác. Dự án Jules, trước đây được Google phát triển, nhằm mục đích mang lại trải nghiệm coding trực quan và biểu cảm hơn cho các developer. Với Stitch, Google tiếp tục khám phá những khả năng của vibe coding. Chi tiết của dự án chưa được tiết lộ đầy đủ, nhưng rõ ràng Google cam kết thúc đẩy lĩnh vực creative coding. Những nỗ lực của công ty trong lĩnh vực này cho thấy sự quan tâm ngày càng tăng đối với sự giao thoa giữa công nghệ và nghệ thuật. Khi ngành công nghệ tiếp tục phát triển, sẽ rất thú vị để xem dự án Stitch của Google phát triển như thế nào và nó có thể ảnh hưởng đến cộng đồng coding rộng lớn hơn ra sao. Thông tin thêm về Stitch dự kiến sẽ được công bố trong tương lai, nhưng hiện tại, dự án vẫn còn là một bí ẩn.

3

Agent Factory: Kết nối các agent, app và dữ liệu bằng các tiêu chuẩn mở mới như MCP và A2A

Agent Factory: Connecting agents, apps, and data with new open standards like MCP and A2A

Microsoft Azure Đọc bài viết →

> Microsoft đã giới thiệu Agent Factory, một nền tảng được thiết kế để kết nối các agent, app và dữ liệu bằng cách sử dụng các tiêu chuẩn mở mới. Nền tảng này tận dụng các tiêu chuẩn Cloud PC (MCP) và Application to Application (A2A) của Microsoft để cho phép tương tác liền mạch giữa các hệ thống và ứng dụng khác nhau. Agent Factory cho phép các developer tạo ra các custom agent có thể tương tác với nhiều ứng dụng và nguồn dữ liệu khác nhau, tạo điều kiện thuận lợi cho việc trao đổi thông tin và tự động hóa các tác vụ. Điều này giúp các doanh nghiệp xây dựng các workflow tích hợp và hiệu quả hơn, cải thiện năng suất và giảm lỗi thủ công. Việc sử dụng các tiêu chuẩn mở như MCP và A2A đảm bảo rằng Agent Factory là platform-agnostic, cho phép các developer tạo ra các agent có thể hoạt động trên các môi trường và hệ sinh thái khác nhau. Bằng cách cung cấp một cách thức tiêu chuẩn hóa để kết nối các agent, app và dữ liệu, Agent Factory nhằm mục đích đơn giản hóa quá trình xây dựng các hệ thống và ứng dụng tích hợp, cuối cùng thúc đẩy đổi mới và tăng trưởng trong nhiều ngành công nghiệp.

4

MCP trên Code Mode

MCP on Code Mode

Changelog Đọc bài viết →

> Matt Carey từ Cloudflare thảo luận về Code Mode và mối quan hệ của nó với MCP (Model-Code-Protocol) trong một tập gần đây. Carey tiết lộ rằng hầu hết mọi người đã hiểu sai về MCP. Anh giải thích cách Code Mode phía server cho phép một MCP server duy nhất expose tất cả 2.500 Cloudflare API endpoint chỉ với khoảng 1.000 token context. Ngoài ra, Carey còn nói về dynamic Worker loader chạy an toàn code do model viết trong một V8 isolate. Cuộc trò chuyện cũng đề cập đến workflow cá nhân của Carey với Claude, vai trò của memory trong tương lai của các agent, và việc anh sử dụng một git wrapper tên là Zaggy để ngăn chặn force-pushing các repository của mình.

5

Sundar Pichai đối mặt với phản đối và cuộc bỏ ra ngoài tại lễ tốt nghiệp Stanford vì mối quan hệ của Google với Israel và ICE

Sundar Pichai faces boos, walkout at Stanford graduation ceremony over Google's Israel, ICE ties

TechCrunch AI Đọc bài viết →

> CEO Google Sundar Pichai đã đối mặt với một cuộc biểu tình và bỏ ra ngoài trong bài phát biểu khai giảng của mình tại Đại học Stanford, nơi ông đã lấy bằng sau đại học. Khoảng 200 sinh viên từ khóa tốt nghiệp đã bỏ ra ngoài, trong khi những người khác la ó ông, để phản đối mối quan hệ quốc phòng của Google và mối quan hệ với cơ quan Thực thi Di trú và Hải quan Hoa Kỳ (ICE). Cuộc biểu tình tập trung vào hợp đồng trị giá 1,2 tỷ USD của Google với Amazon để cung cấp các dịch vụ cloud và AI cho quân đội Israel, cũng như mối quan hệ của họ với ICE. Sinh viên đã giơ các biểu ngữ và vẫy cờ Palestine, bày tỏ sự phản đối việc Google tham gia vào các vấn đề này. Cuộc bỏ ra ngoài được tổ chức bởi các nhóm hoạt động trong khuôn viên trường, bao gồm Stanford Students for Justice in Palestine và No Tech for Apartheid. Sự việc này đánh dấu sự kiện mới nhất trong một loạt các cuộc biểu tình và chỉ trích chống lại việc Google tham gia vào Project Nimbus và mối quan hệ của họ với ICE.

6

datasette-agent 0.3a0

datasette-agent 0.3a0

Simon Willison Đọc bài viết →

> Một phiên bản mới của datasette-agent, 0.3a0, đã được phát hành. Bản cập nhật này bao gồm một cơ chế user approval, cho phép người dùng được nhắc nhở trước khi thực hiện một số operation nhất định. Công cụ "execute_write_sql" mới giờ đây có thể yêu cầu approval cho nhiều hành động khác nhau, chẳng hạn như thêm dữ liệu vào một table. Chế độ chat terminal của datasette agent cũng đã được cải tiến để hỗ trợ approvals. Ngoài ra, một số tùy chọn mới đã được thêm vào, bao gồm chế độ "--unsafe" tự động approve các operation. Chế độ này

7

Profiling trong PyTorch (Phần 2): Từ nn.Linear đến một Fused MLP

Profiling in PyTorch (Part 2): From nn.Linear to a Fused MLP

Hugging Face Blog Đọc bài viết →

Trong bài viết này, tác giả tiếp tục khám phá việc phân tích hiệu suất của PyTorch, tập trung vào mô-đun nn.Linear và các hoạt động cơ bản của nó. Tác giả thay thế việc nhân ma trận và cộng viết tay bằng mô-đun nn.Linear, là một khối xây dựng phổ biến trong các model học sâu. Mô-đun này sau đó được xếp chồng với các mô-đun nn.Linear khác và một hàm kích hoạt để tạo thành một khối Multilayer Perceptron (MLP). Tác giả kiểm tra dấu vết của bộ phân tích hiệu suất trong một cuộc gọi forward của lớp tuyến tính và lưu ý rằng hoạt động chuyển vị được sử dụng để viết lại siêu dữ liệu tensor, thay vì khởi chạy một kernel trên GPU. Việc thêm bias cũng được tích hợp vào kernel nhân ma trận bằng cách sử dụng một epilogue, giúp tránh tải hoặc ghi vào HBM lần thứ hai. Tác giả cũng thảo luận về việc sử dụng torch.compile và sự khác biệt giữa hai loại kernel GEMM. Ngoài ra, họ giới thiệu khái niệm về các kernel được tinh chỉnh thủ công và thư viện kernel, có thể cung cấp hiệu suất tốt hơn. Bài viết kết thúc bằng việc nhấn mạnh lợi ích của việc sử dụng các kernel đã được tinh chỉnh.

8

Cách biến Tweets thành Video Viral bằng AI: T3P Agent Framework

How to Turn Tweets Into Viral Videos With AI: The T3P Agent Framework

Dev.to AI Đọc bài viết →

Các nhà sáng tạo công nghệ đang gặp khó khăn trong việc chuyển đổi các tweet thành video lan truyền do quá trình chỉnh sửa thủ công tốn thời gian. Quy trình làm việc trung bình mất khoảng 4-6 giờ, chiếm 33% thời gian lan truyền 18 giờ của tweet. Tuy nhiên, nghiên cứu cho thấy thời gian là rất quan trọng, với 7/10 video được xuất bản trong 3 giờ đầu vượt trội so với video 10/10 được xuất bản 9 giờ sau. Để vượt qua thách thức này, một khuôn khổ mới gọi là T3P Agent Framework đã được phát triển. Khuôn khổ này sử dụng trí tuệ nhân tạo (AI) để tự động hóa quá trình chuyển đổi tweet thành video lan truyền. Nó kết hợp nhiều công cụ, bao gồm n8n cho việc dàn xếp, LangGraph cho logic tác nhân trạng thái, GPT-4o cho kịch bản, ElevenLabs cho giọng nói và Runway Gen-3 cho cảnh quay. Khuôn khổ T3P Agent Framework thu gọn quy trình làm việc thủ công thành một tác nhân không cần giám sát, xuất bản video trước khi thời gian lan truyền kết thúc. Bằng cách loại bỏ con người khỏi giữa các giai đoạn, các nhà sáng tạo có thể giảm chi phí thời gian cho mỗi video xuống một bậc. Khuôn khổ này có sẵn với giá dưới 47 đô la/tháng và có thể được xây dựng bởi bất kỳ ai, giúp nó trở nên dễ tiếp cận với các nhà sáng tạo

> muốn chuyển đổi tweet thành video lan truyền với sự hỗ trợ của LLM và API, cũng như các nhà phát triển muốn tích hợp framework này vào model của họ thông qua token và các công cụ khác.

TIPS & TRICKS CHO DEV

Cài đặt Ollama

Vấn đề: Chạy Local LLM trên máy cá nhân mà không cần internet.
Cách làm: Sử dụng lệnh `pip install ollama` để cài đặt, sau đó chạy `ollama --model lmstudio` để khởi động. Ví dụ, chạy lệnh `ollama --model lmstudio --prompt "Tạo một đoạn văn về chủ đề AI"` để tạo văn bản.
Đánh giá: Hiệu quả cho dự án cần sự riêng tư, không nên dùng khi máy có tài nguyên thấp.

Tối ưu LM Studio

Vấn đề: Tối ưu hiệu suất của Local LLM khi chạy trên máy cá nhân.
Cách làm: Sử dụng lệnh `lmstudio --optimize` để tối ưu hóa, sau đó chạy `lmstudio --model optimize` để áp dụng. Ví dụ, chạy lệnh `lmstudio --optimize --model lmstudio` để tối ưu hóa mô hình.
Đánh giá: Hiệu quả khi máy có tài nguyên hạn chế, nên dùng thường xuyên.

Sử dụng CLI Ollama

Vấn đề: Sử dụng dòng lệnh để chạy Local LLM trên máy cá nhân.
Cách làm: Sử dụng lệnh `ollama --help` để xem hướng dẫn, sau đó chạy `ollama --model lmstudio --prompt "Tạo một đoạn văn"` để tạo văn bản. Ví dụ, chạy lệnh `ollama --model lmstudio --prompt "Tạo một đoạn văn về chủ đề AI"` để tạo văn bản.
Đánh giá: Hiệu quả khi cần chạy nhanh, nên dùng khi cần tạo văn bản nhanh chóng.

BÀI HỌC AI HÔM NAY CHO DEV

1. **Tối ưu chi phí & hiệu năng LLM**
2. Dev cần biết cách tối ưu chi phí và hiệu năng của mô hình ngôn ngữ lớn (LLM) để đảm bảo ứng dụng AI của họ hoạt động hiệu quả và tiết kiệm. Điều này giúp giảm thiểu chi phí vận hành và nâng cao trải nghiệm người dùng. Ngoài ra, tối ưu hóa LLM cũng giúp giảm thiểu tác động môi trường của các ứng dụng AI.
3. Ví dụ, sử dụng kỹ thuật fine-tuning và LoRA (Low-Rank Adaptation) có thể giúp giảm kích thước mô hình và tăng tốc độ xử lý, từ đó giảm chi phí và cải thiện hiệu năng.
4. Tip hoặc bước tiếp theo: Sử dụng các thư viện như Hugging Face

Transformers để tối ưu hóa LLM và thử nghiệm với các kỹ thuật khác nhau để tìm ra giải pháp phù hợp cho ứng dụng của bạn.

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

Nguồn: Google News · Groq AI