BIS - MT · 20/06/2026

Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

> *“Your time is limited, so don't waste it living someone else's life.”*
>
> ↪ Thời gian của bạn có hạn, vì vậy đừng lãng phí nó để sống cuộc đời của người khác.
>
> — Steve Jobs

Cuộc đời quá ngắn để sống theo kỳ vọng của người khác — hãy dũng cảm theo đuổi con đường của chính mình.

TIN TỨC NỔI BẬT

1 Claude Code vs GitHub Copilot 2026: SWE-bench, Giá cả [Thử nghiệm thực tế]

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 kiểm tra về hiệu suất và giá cả của chúng. Công cụ SWE-bench, một công cụ đánh giá hiệu suất cho các kỹ sư phần mềm, đã được sử dụng để đánh giá hai trợ lý mã hóa được hỗ trợ bởi AI này. Kết quả cho thấy Claude Code vượt trội so với GitHub Copilot trong một số lĩnh vực chính, bao gồm hoàn thành mã và xem xét mã. Tuy nhiên, GitHub Copilot được tìm thấy là chính xác hơn trong một số kịch bản nhất định, đặc biệt là khi nói đến đề xuất mã. Khi nói đến giá cả, Claude Code cung cấp một mô hình giá cả linh hoạt hơn, với một kế hoạch miễn phí có sẵn cho các cá nhân và một hệ thống giá cấp bậc cho các nhóm. Ngược lại, GitHub Copilot yêu cầu một đăng ký GitHub, điều này có thể là một chi phí đáng kể cho các nhóm lớn. Tổng thể, so sánh này nhấn mạnh điểm mạnh và điểm yếu của từng công cụ, và có thể giúp các nhà phát triển đưa ra quyết định thông minh khi chọn giữa Claude Code và GitHub Copilot cho nhu cầu mã hóa của họ.

2 Agent Factory: Kết nối các agent, app và data bằng các open standard 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 tác nhân, ứng dụng và dữ liệu bằng cách sử dụng các tiêu chuẩn mở. Nền tảng này tận dụng các tiêu chuẩn Cloud PC (MCP) và Application to Application (A2A) của Microsoft để tạo điều kiện cho sự tương tác liền mạch giữa các thực thể khác nhau. Agent Factory cho phép các nhà phát triển xây dựng và triển khai các tác nhân thông minh có thể tương tác với các ứng dụng và nguồn dữ liệu khác nhau. Điều này cho phép tạo ra các hệ thống phức tạp và tích hợp hơn, nơi các tác nhân có thể thực hiện các nhiệm vụ tự động và đưa ra quyết định dựa trên dữ liệu thời gian thực. Việc sử dụng các tiêu chuẩn MCP và A2A đảm bảo khả năng tương tác và tương thích trên các nền tảng khác nhau, giúp các nhà phát triển dễ dàng tích hợp các ứng dụng và nguồn dữ liệu của họ. Bằng cách cung cấp một khuôn khổ tiêu chuẩn hóa cho các tương tác của tác nhân, Agent Factory nhằm mục đích đẩy nhanh sự phát triển của các hệ thống và ứng dụng thông minh có thể học hỏi, thích nghi và cải thiện theo thời gian. Công nghệ này có tiềm năng thay đổi các ngành công nghiệp khác nhau, bao gồm chăm sóc sức khỏe, tài chính và dịch vụ khách hàng, bằng cách tận dụng các công nghệ như AI, API, LLM, model, token, và framework.

3 AWS open-source MCP Server cho Bedrock AgentCore để tinh gọn phát triển AI Agent

AWS Open-Sources an MCP Server for Bedrock AgentCore to Streamline AI Agent Development

MarkTechPost Đọc bài viết →

Amazon Web Services (AWS) đã mở mã nguồn một máy chủ MCP (Nền tảng Đa đám mây) cho Bedrock AgentCore, một framework để phát triển các tác nhân AI. Động thái này nhằm mục đích đơn giản hóa quá trình phát triển tác nhân AI bằng cách cung cấp một nền tảng tiêu chuẩn hóa và mã nguồn mở. Máy chủ MCP được thiết kế để hoạt động với Bedrock AgentCore, cho phép các nhà phát triển xây dựng, triển khai và quản lý các tác nhân AI trên nhiều nền tảng đám mây khác nhau. Bằng cách mở mã nguồn máy chủ MCP, AWS đang cung cấp một nền tảng được thúc đẩy bởi cộng đồng cho các nhà phát triển đóng góp và cải thiện framework. Điều này có thể dẫn đến sự đổi mới nhanh hơn và sự hợp tác tốt hơn giữa các nhà phát triển. Máy chủ MCP mã nguồn mở dự kiến sẽ mang lại lợi ích cho cộng đồng phát triển AI rộng

lớn hơn bằng cách giảm độ phức tạp của việc phát triển và triển khai các tác nhân AI. Nó cũng sẽ cho phép các nhà phát triển tận dụng điểm mạnh của các nền tảng đám mây khác nhau, cuối cùng dẫn đến việc phát triển tác nhân AI hiệu quả và hiệu quả hơn. Động thái này được coi là một bước tiến để dân chủ hóa thêm việc phát triển AI và làm cho nó trở nên dễ tiếp cận hơn với nhiều nhà phát triển.

4

MosaicLeaks: Liệu research agent của bạn có giữ được bí mật?

MosaicLeaks: Can your research agent keep a secret?

Hugging Face Blog Đọc bài viết →

Các nhà nghiên cứu đã xác định một rủi ro riêng tư đáng kể trong các tác nhân nghiên cứu sâu, kết hợp tài liệu cục bộ riêng tư với các công cụ bên ngoài như thu thập thông tin từ web. Điều này có thể dẫn đến thông tin nhạy cảm bị rò rỉ thông qua các truy vấn bên ngoài của tác nhân. Một nghiên cứu mới, MosaicLeaks, đề xuất một nhiệm vụ nghiên cứu sâu mới với các câu hỏi đa bước xen kẽ thông tin công khai và riêng tư để kiểm tra rủi ro này. Nghiên cứu cho thấy các tác nhân thường xuyên rò rỉ thông tin riêng tư, và việc đào tạo chỉ để thực hiện nhiệm vụ làm cho tình hình trở nên tồi tệ hơn. Để giải quyết vấn đề này, các nhà nghiên cứu đề xuất một phương pháp đào tạo RL nhận thức về rò rỉ mozaic, Nghiên cứu Sâu Nhận thức Riêng tư (PA-DR). Phương pháp này làm tăng tỷ lệ thành công của các tác nhân trong việc trả lời câu hỏi chính xác đồng thời giảm thiểu việc rò rỉ thông tin riêng tư. Nghiên cứu chứng minh hiệu quả của PA-DR bằng cách so sánh hiệu suất của nó với một mô hình cơ sở. Các nhà nghiên cứu đã tạo ra một bộ dữ liệu, MosaicLeaks, chứa 1.001 chuỗi nghiên cứu đa bước trên tài liệu doanh nghiệp cục bộ và một tập hợp web được kiểm soát. Mục tiêu là tạo ra các nhiệm vụ có thể gây ra rò rỉ riêng tư từ tài liệu doanh nghiệp trong khi vẫn có thể giải quyết được mà không bị rò rỉ. Nghiên cứu nhấn mạnh tầm quan trọng của việc thiết kế các tác nhân nghiên cứu có thể tìm kiếm an toàn và tránh rò rỉ thông tin nhạy cảm.

5

Giới thiệu LangSmith’s No Code Agent Builder

Introducing LangSmith’s No Code Agent Builder

LangChain Blog Đọc bài viết →

LangSmith đã giới thiệu một công cụ mới gọi là No Code Agent Builder, được thiết kế để giúp những người không phải là nhà phát triển dễ dàng xây dựng các agent hơn. Trải nghiệm không cần mã này cho phép người dùng tạo agent với bộ nhớ và tạo lời nhắc được hướng dẫn, giảm rào cản gia nhập để xây dựng các ứng dụng có tính chất agent. Agent Builder là một phần của nền tảng kỹ thuật agent của LangSmith, giúp các nhà phát triển gỡ lỗi và triển khai agent. Không giống như các công cụ xây dựng workflow trực quan, Agent Builder của LangSmith tập trung vào việc xây dựng agent từ bốn thành phần cốt lõi, cho phép người dùng giải quyết các nhiệm vụ phức tạp. Công cụ này phù hợp cho các trường hợp sử dụng năng suất nội bộ, chẳng hạn như trợ lý email và trò chuyện, và có thể được sử dụng để tự động hóa các nhiệm vụ như gửi tóm tắt lịch hoặc tạo bước tiếp theo dựa trên tin nhắn. LangSmith đã tích hợp các bài học từ các framework mã nguồn mở và các phiên bản đầu tiên của sản phẩm để thông báo các quyết định thiết kế của mình. Agent Builder hiện đang trong giai đoạn xem trước riêng tư và có thể được thử nghiệm bởi những người dùng quan tâm, với công ty mong đợi nhận được phản hồi từ cộng đồng để tiếp tục cải thiện trải nghiệm.

6

7.000 Langflow server bị tấn công. LangGraph và LangChain cũng có lỗ hổng tương tự

7,000 Langflow servers are under attack. LangGraph and LangChain have the same holes

VentureBeat Đọc bài viết →

Ba khuôn khổ đại lý AI được sử dụng rộng rãi, LangGraph, Langflow và LangChain, đã được phát hiện có các lỗ hổng quan trọng có thể bị các kẻ tấn công khai thác. LangGraph và LangChain có cùng các lỗ hổng, được phát hiện bởi các nhà nghiên cứu khác nhau. Các lỗ hổng này cho phép các kẻ tấn công thực hiện mã từ xa (RCE) và truy cập dữ liệu nhạy cảm, bao gồm cả khóa OpenAI và thông tin đăng nhập cơ sở dữ liệu. LangGraph có lỗ hổng tiêm SQL (CVE-2025-67644) có thể được kết hợp với một lỗ hổng khác (CVE-2026-28277) để đạt được RCE. LangChain có lỗ hổng traversal đường dẫn (CVE-2026-34070) có thể được kết hợp với một lỗi deserialization (CVE-2025-68664) để đọc các tệp nhạy cảm, bao gồm cả khóa API. Langflow có lỗ hổng traversal đường dẫn (CVE-2026-5027) có thể bị khai thác bởi một kẻ tấn công để viết một tệp vào máy chủ đại lý và đạt được shell. Lỗ hổng này đã bị khai thác tích cực trong tự nhiên, với hơn 7.000 trường hợp□□ được xác định. Các nhà nghiên cứu nhấn mạnh rằng những lỗ hổng này

không phải là những lỗ hổng kỳ lạ hoặc cụ thể cho AI, mà là những lỗi AppSec cổ điển đã có mặt trong các khuôn khổ trong một thời gian. Nguyên nhân gốc rễ của vấn đề là các thiết lập mặc định không an toàn và thiếu xác thực và đặc quyền tối thiểu trong các khuôn khổ. Để giải quyết những lỗ hổng này, các đội an ninh nên tuân theo một danh sách kiểm tra gồm sáu câu hỏi để xác minh các ranh giới tin cậy của khuôn khổ đại lý AI của họ. Danh sách kiểm tra bao gồm các câu hỏi như liệu kho lưu trữ trạng thái của đại lý có thể bị đầu độc bằng mã, liệu một yêu cầu không xác thực có thể viết một tệp vào máy chủ đại lý, và liệu các khuôn khổ có đang chạy ngoài quản lý an ninh. Các bản vá là các bản cập nhật phiên bản và thay đổi cấu hình có thể được thực hiện trong tuần này, và các đội an ninh nên ưu tiên vá lỗ hổng khi công bố, không chờ đợi danh mục liên bang.

7

Cách chúng tôi xây dựng một internal data analytics agent

How we built an internal data analytics agent

GitHub Blog Đọc bài viết →

GitHub, một nền tảng hàng đầu dành cho developer, đã phát triển một tác nhân phân tích dữ liệu nội bộ gọi là Qubot. Qubot được hỗ trợ bởi GitHub Copilot, một công cụ tạo mã code AI, và cho phép bất kỳ nhân viên GitHub nào đặt câu hỏi về dữ liệu của công ty bằng ngôn ngữ tự nhiên. Mục tiêu của Qubot là cung cấp quyền truy cập tự phục vụ vào dữ liệu và thông tin, giúp các đội sản phẩm đưa ra quyết định dễ dàng hơn. Ở quy mô của GitHub, việc cung cấp hỗ trợ phân tích chuyên dụng cho nhiều đội là một thách thức, và Qubot nhằm lấp đầy khoảng trống này. Với Qubot, nhân viên có thể truy vấn các model dữ liệu, bộ lọc và kết quả mà không cần hỗ trợ của một nhà phân tích dữ liệu. Đổi mới này là một phần trong nỗ lực của GitHub để tận dụng trí tuệ nhân tạo và học máy để cải thiện trải nghiệm của developer và làm cho dữ liệu trở nên dễ tiếp cận hơn.

8

Tính năng usage analytics mới và cập nhật spend control cho doanh nghiệp

New usage analytics and updated spend controls for enterprises

OpenAI Blog Đọc bài viết →

OpenAI đã giới thiệu các tính năng mới cho nền tảng ChatGPT Enterprise của mình. Công ty đã thêm phân tích sử dụng và cập nhật

> kiểm soát chi tiêu để giúp các tổ chức quản lý chi phí và mở rộng việc sử dụng trí tuệ nhân tạo (AI) một cách hiệu quả hơn. Các tính năng mới này nhằm cung cấp cho các doanh nghiệp sự tự tin hơn trong việc triển khai AI, cho phép họ hiểu rõ hơn về cách nhân viên của họ sử dụng nền tảng và đưa ra quyết định thông minh hơn về chi tiêu của mình. Kiểm soát chi tiêu cập nhật sẽ cho phép các tổ chức đặt giới hạn và theo dõi sử dụng, trong khi phân tích sử dụng sẽ cung cấp thông tin chi tiết về cách ChatGPT Enterprise được sử dụng trong công ty của họ. Động thái này có thể sẽ được các doanh nghiệp chào đón, những doanh nghiệp đang tìm cách tích hợp AI vào hoạt động của mình, vì nó sẽ giúp họ tối ưu hóa việc sử dụng công nghệ và giảm thiểu chi phí không cần thiết. Các tính năng mới là một phần của nỗ lực liên tục của OpenAI để hỗ trợ nhu cầu ngày càng tăng về các giải pháp AI trong lĩnh vực doanh nghiệp.

TIPS & TRICKS CHO DEV

Tạo mã nhanh chóng

Vấn đề: Viết mã từ đầu tốn nhiều thời gian và công sức.

Cách làm: Sử dụng Claude Code với lệnh `claude code` và cung cấp prompt như "write a Python function to sort a list".

Đánh giá: Hiệu quả khi cần viết nhanh, nhưng nên kiểm tra lại mã.

Debug mã tự động

Vấn đề: Debug mã động tốn nhiều thời gian.

Cách làm: Chạy lệnh `claude debug` với mã có lỗi và Claude Code sẽ tự động tìm kiếm và đề xuất sửa lỗi.

Đánh giá: Rất hiệu quả khi gặp lỗi phức tạp, tiết kiệm thời gian.

Tối ưu mã hiện có

Vấn đề: Mã hiện có không tối ưu và cần cải thiện.

Cách làm: Sử dụng lệnh `claude refactor` và cung cấp mã cần tối ưu hóa.

Đánh giá: Giúp cải thiện hiệu suất và bảo trì mã, nên dùng thường xuyên.

BÀI HỌC AI HÔM NAY CHO DEV

1. **Tối ưu chi phí & hiệu năng LLM**
2. Để tối ưu hóa chi phí và hiệu năng của mô hình ngôn ngữ lớn (LLM), các nhà phát triển cần biết cách giảm thiểu tài nguyên tính toán và tăng tốc độ xử lý. Điều này đặc biệt quan trọng khi triển khai mô hình trên các thiết bị edge hoặc trong các ứng dụng thời gian thực.
3. Ví dụ, có thể sử dụng kỹ thuật fine-tuning và LoRA (Low-Rank Adaptation) để

giảm kích thước mô hình và tăng tốc độ huấn luyện. Điều này có thể được thực hiện bằng cách tải mô hình pre-trained và điều chỉnh các lớp trên cùng để phù hợp với nhiệm vụ cụ thể.

4. Tip hoặc bước tiếp theo: Nghiên cứu và áp dụng các kỹ thuật như quantization, pruning và knowledge distillation để tối ưu hóa mô hình LLM và giảm chi phí tính toán.