BIS - MT · 08/06/2026

Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

> *"The only impossible journey is the one you never begin."*
>
> ↪ Hành trình duy nhất không thể thực hiện là hành trình bạn chưa bao giờ bắt đầu.
>
> — Tony Robbins

Mọi thành tựu đều bắt đầu từ quyết định đầu tiên — điều duy nhất đảm bảo thất bại là không bao giờ thử.

TIN TỨC NỔI BẬT

1

So sánh 12 Trợ lý Lập trình AI năm 2026: Claude Code vs Antigravity vs Codex vs Cursor vs OpenCode vs Hermes

12 AI Coding Agents Compared in 2026: Claude Code vs Antigravity vs Codex vs Cursor vs OpenCode vs Hermes

Security Boulevard Đọc bài viết →

> Trong một so sánh gần đây, sáu tác nhân mã hóa AI đã được đánh giá vào năm 2026. Các tác nhân được thử nghiệm bao gồm Claude Code, Antigravity, Codex, Cursor, OpenCode và Hermes. So sánh này nhằm mục đích đánh giá khả năng và hiệu suất của các công cụ mã hóa được hỗ trợ bởi AI. Đánh giá liên quan đến việc kiểm tra khả năng mã hóa của các tác nhân, bao gồm khả năng hiểu và tạo mã, cũng như khả năng gỡ lỗi và tối ưu hóa mã hiện có. Kết quả cho thấy mỗi tác nhân có điểm mạnh và điểm yếu, và không có tác nhân nào vượt trội trong tất cả các lĩnh vực. Claude Code, được phát triển bởi Google, đã thể hiện khả năng tạo mã mạnh mẽ, trong khi Antigravity đã thể hiện kỹ năng gỡ lỗi ấn tượng. Codex, một sản phẩm của Microsoft, đã hoạt động tốt trong việc hiểu và tạo mã. Cursor, OpenCode và Hermes cũng đã thể hiện khả năng đáng chú ý, nhưng với mức độ thành công khác nhau. So sánh này nhấn mạnh sự đa dạng của các tác nhân mã hóa AI có sẵn và nhu cầu của các nhà phát triển để chọn công cụ phù hợp nhất với nhu cầu cụ thể của họ.

2

Claude Code vs GitHub Copilot 2026: SWE-bench, Giá cả [Đã Kiểm tra]

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 thử nghiệm về hiệu suất và giá cả của chúng. Cả hai công cụ mã hóa được hỗ trợ bởi AI đều nhằm mục đích hỗ trợ các kỹ sư phần mềm trong công việc của họ, nhưng chúng khác nhau về cách tiếp cận và khả năng. Công cụ SWE-bench, một công cụ đánh giá hiệu suất cho 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 trong một số lĩnh vực, bao gồm hoàn thành mã, xem xét mã và tạo mã. Tuy nhiên, GitHub Copilot được tìm thấy là hiệu quả hơn trong một số kịch bản nhất định, chẳng hạn như tóm tắt mã và dịch mã. Về giá cả, GitHub Copilot cung cấp một kế hoạch miễn phí, cũng như một đăng ký trả phí bắt đầu từ 10 đô la mỗi tháng. Claude Code, mặt khác, cung cấp một thử nghiệm miễn phí, nhưng kế hoạch giá của nó không được nêu rõ trong so sánh. Kết quả của thử nghiệm gợi ý rằng Claude Code có thể là một lựa chọn hiệu quả hơn cho các kỹ sư phần mềm, nhưng sự khác biệt về giá có thể là một yếu tố quyết định cho một số người dùng.

3

Giới thiệu Thông số kỹ thuật Trợ lý Mở (Agent Spec): Một Đại diện Thống nhất cho Trợ lý AI

Introducing the Open Agent Specification (Agent Spec): A Unified Representation for AI Agents

Oracle Blogs Đọc bài viết →

> Oracle đã giới thiệu Open Agent Specification (Agent Spec), một biểu diễn thống nhất cho các tác nhân AI. Agent Spec nhằm cung cấp một khuôn khổ tiêu chuẩn hóa để mô tả và tương tác với các tác nhân AI, cho phép tích hợp và giao tiếp liền mạch trên các hệ thống và nền tảng khác nhau. Tiêu chuẩn này phác thảo một cấu trúc chung cho các tác nhân AI, bao gồm mục tiêu, khả năng và hành vi của chúng. Biểu diễn tiêu chuẩn hóa này được thiết kế để thúc đẩy sự phát triển của các hệ thống AI tinh vi và tương tác hơn. Bằng cách áp dụng Agent Spec, các nhà phát triển có thể tạo ra các tác nhân AI có thể tương tác dễ dàng với các hệ thống khác, bất kể kiến trúc cơ bản hoặc triển khai của chúng. Điều này, đến lượt, có thể dẫn đến các ứng dụng AI hiệu quả và hiệu suất cao hơn trên nhiều ngành công nghiệp khác nhau. Agent Spec là một tiêu chuẩn mở, cho phép các nhà phát triển đóng

góp vào sự phát triển của nó và đảm bảo sự liên tục phù hợp và thích ứng với các công nghệ AI mới nổi. Với Agent Spec, Oracle nhằm mục đích thúc đẩy một hệ sinh thái AI thống nhất và hợp tác hơn, nơi các hệ thống và nền tảng khác nhau có thể làm việc cùng nhau một cách liền mạch.

4

Từ các dự án mã nguồn mở thành OpenAI

From open source hits to OpenAI

Changelog Đọc bài viết →

Trong một cuộc trò chuyện gần đây, Max Stoiber, một developer tại OpenAI, đã thảo luận về các dự án mã nguồn mở và tác động của chúng đối với ngành công nghiệp công nghệ. Stoiber đã đề cập đến tầm quan trọng của các dự án mã nguồn mở ít được biết đến, chẳng hạn như react-boilerplate và styled-components, đã đóng góp đáng kể cho ngành công nghiệp. Ông cũng nhắc đến việc GitHub mua lại Spectrum và vai trò của nó trong việc định hình GitHub Discussions. Ngoài ra, Stoiber đã nói về sự phát triển của Stellate, một bộ nhớ đệm GraphQL được Shopify và The Guild mua lại. Hơn nữa, ông đã thảo luận về sự phát triển của các ứng dụng ChatGPT, mà ông tin rằng cung cấp một bề mặt mới cho phát triển phần mềm.

5

Dreaming: Bộ nhớ tốt hơn cho ChatGPT hữu ích hơn

Dreaming: Better memory for a more helpful ChatGPT

OpenAI Blog Đọc bài viết →

ChatGPT đã giới thiệu một hệ thống bộ nhớ mới được thiết kế để cải thiện khả năng nhớ lại sở thích của người dùng và duy trì ngữ cảnh trong suốt cuộc trò chuyện. Nâng cấp này nhằm mục đích làm cho chatbot trở nên hữu ích và hiệu quả hơn trong các tương tác của nó. Hệ thống bộ nhớ mới này được thiết kế để nhớ tốt hơn sở thích của người dùng, cho phép đưa ra các phản hồi được cá nhân hóa và phù hợp hơn. Việc cải tiến này dự kiến sẽ nâng cao trải nghiệm người dùng tổng thể, làm cho ChatGPT trở thành một công cụ đáng tin cậy và hiệu quả hơn cho người dùng. Bằng cách giữ cho ngữ cảnh luôn mới mẻ và phù hợp, chatbot sẽ có thể cung cấp thông tin chính xác và hữu ích hơn, cuối cùng làm cho nó trở thành một tài nguyên quý giá hơn cho những người tương tác với nó. Việc giới thiệu hệ thống bộ nhớ mới này là một bước tiến quan trọng trong sự phát triển của ChatGPT, và sẽ rất

thú vị khi xem nó ảnh hưởng đến trải nghiệm người dùng như thế nào trong tương lai.

6 Agentic AI giải quyết việc lập trình — và□lộ mọi vấn đề khác trong kỹ thuật phần mềm

Agentic AI solved coding — and exposed every other problem in software engineering

VentureBeat Đọc bài viết →

Agentic AI đã cách mạng hóa quá trình kỹ thuật, cho phép tạo và thực thi mã nhanh hơn. Tuy nhiên, mặc dù tốc độ tăng lên, nhưng cải tiến sản phẩm vẫn chưa theo kịp. Lý do nằm ở sự phức tạp của việc định nghĩa yêu cầu, tích hợp với hệ thống và duy trì phần mềm trong điều kiện thế giới thực. Khi mã được AI tạo ra tăng quy mô, việc xem xét của con người đã trở thành một nút thắt cổ chai, và các kỹ sư gặp khó khăn trong việc phát hiện sai lầm của agent. Để điều hướng□□ này, các nhà lãnh đạo kỹ thuật doanh nghiệp phải phát triển một cuốn sách chơi có chủ đích. Giai đoạn đầu tập trung vào quản trị tài chính và rủi ro, bao gồm bảo mật cơ sở hạ tầng, hạn chế chảy máu tài chính và thực thi quản trị như một rủi ro hàng đầu. Điều này liên quan đến việc đối xử với cấu hình agent như cơ sở hạ tầng sản xuất, thực thi đặc quyền tối thiểu cho các actor không phải con người và triển khai sự tách biệt nghiêm ngặt giữa quyền truy cập đọc và ghi / thực thi. Giai đoạn thứ hai liên quan đến việc xây dựng một chiến lược kỹ thuật, bao gồm việc chọn các model phù hợp, đo lường thành công của chúng và triển khai đa model và đa nhà cung cấp. Cách tiếp cận này cho phép các tổ chức đặc tả chính xác hành vi và ranh giới hiệu suất trên các model và hiểu nơi mỗi model vượt trội. Giai đoạn thứ ba tập trung vào tài năng và tổ chức, tái định hướng vốn nhân lực để quản lý nút thắt cổ chai mới. Điều này liên quan đến việc nâng cao kỹ năng cho các kỹ sư để chuyển từ những người viết cú pháp sang những người nghĩ hệ thống và quản lý agent, định lại hiệu suất và khuyến khích để thưởng cho tác động kinh doanh mở rộng, và tránh cắt giảm nhân sự trước khi thích nghi với chiến lược mới, bao gồm cả việc sử dụng AI, API, LLM, model, token, developer, framework, v.v.

7 Tải xuống: Hack AI vượt qua Mythos, và tác động của rô-bốt trò chuyện đối với não bộ chúng ta

The Download: AI hacking beyond Mythos, and chatbots' impact on our brains

MIT Tech Review Đọc bài viết →

Trong tin công nghệ ngày hôm nay, lo ngại về việc hack AI đã trở lại sau khi những kẻ tấn công khai thác trình đại lý hỗ trợ khách hàng AI của Meta để đánh cắp tài khoản Instagram. Sự việc này làm nổi bật rủi ro của các cuộc tấn công đơn giản gây ra thiệt hại, ngay cả khi các công ty chuyển nhiều công việc hơn cho AI. Thông báo của Anthropic rằng mô hình Mythos của họ quá tốt trong việc hack để phát hành chung đã gây ra lo ngại về an ninh mạng về các hệ thống AI siêu mạnh gây quá tải cho cơ sở hạ tầng máy tính. Trong khi đó, nhà tâm lý học Gloria Mark cảnh báo rằng việc quá phụ thuộc vào các công cụ AI như ChatGPT và Claude có thể làm suy yếu khả năng nhận thức của chúng ta, dẫn đến hiệu suất thấp hơn, căng thẳng cao hơn và sự suy giảm khả năng tư duy phản biện và trí tuệ cảm xúc. Mark tin rằng việc thay đổi mối quan hệ của chúng ta với những công nghệ này có thể giúp giảm thiểu sự thay đổi này. Ngoài ra, sự phát triển AI đang được kêu gọi giảm tốc trên toàn cầu bởi Anthropic, với lý do rủi ro của các mô hình "tự cải tiến".

8

Cung cấp cho trợ lý của bạn một máy tính riêng

Give your agent its own computer

LangChain Blog Đọc bài viết →

Các công ty công nghệ đang phải đối mặt với một thách thức đáng kể trong việc tạo ra các tác nhân trí tuệ nhân tạo (AI) có thể thực thi mã một cách an toàn và hiệu quả. Những tác nhân này yêu cầu một máy tính chuyên dụng với hệ thống tệp, shell và trình quản lý gói để chạy hàng triệu nhiệm vụ, nhưng việc cấp quyền truy cập vào cơ sở hạ tầng của công ty là một rủi ro bảo mật. Satya Nadella của Microsoft đã nhấn mạnh nhu cầu mỗi tác nhân phải có máy tính của riêng nó. Để giải quyết vấn đề này, các công ty đang chuyển sang các môi trường sandbox chuyên dụng, chẳng hạn như LangSmith Sandboxes, cung cấp một không gian an toàn và cách ly để các tác nhân chạy mã. Những sandbox này được thiết kế để là các microVM ảo hóa phần cứng, cung cấp khả năng khởi động tức thời của một hàm không có máy chủ và tính trạng thái của một máy đầy đủ. Điều này cho phép các tác nhân cài đặt gói, chạy kịch bản và chỉnh sửa tệp mà không làm tổn hại đến cơ sở hạ tầng của công ty. LangSmith Sandboxes cũng cung cấp một tập hợp các nguyên thủy giúp các quy trình làm việc của tác nhân sẵn sàng cho sản xuất, bao gồm cả ảnh chụp nhanh và fork, cho phép các nhóm bắt và khởi động các sandbox mới từ ảnh chụp nhanh giữa phiên. Cách tiếp cận này không chỉ cải thiện bảo mật mà

> còn tăng cường hiệu quả, đặc biệt là cho các công việc GPU, bằng cách giảm thiểu thời gian nhàn rỗi và giảm chi phí khi mở rộng quy mô.

TIPS & TRICKS CHO DEV

Quản lý Context Window

Vấn đề: Giới hạn context window gây khó khăn cho việc xử lý văn bản dài.
Cách làm: Sử dụng kỹ thuật chunking, ví dụ: chia văn bản thành các đoạn 2048 tokens để phù hợp với context window của mô hình.
Đánh giá: Hiệu quả khi xử lý văn bản dài, nhưng có thể tăng độ phức tạp của mã nguồn.

Tối ưu hóa Long-Context

Vấn đề: Mô hình AI không thể xử lý văn bản dài do giới hạn bộ nhớ.
Cách làm: Sử dụng kỹ thuật attention mechanism, ví dụ: sử dụng lệnh `--attention-window` trong CLI của mô hình Transformer.
Đánh giá: Hiệu quả khi xử lý văn bản dài, nhưng có thể giảm hiệu suất của mô hình.

Quản lý Memory

Vấn đề: Mô hình AI tiêu tốn quá nhiều bộ nhớ khi xử lý dữ liệu lớn.
Cách làm: Sử dụng kỹ thuật caching, ví dụ: sử dụng thư viện `joblib` để lưu trữ kết quả trung gian.
Đánh giá: Hiệu quả khi xử lý dữ liệu lớn, nhưng có thể tăng thời gian khởi tạo của mô hình.

BÀI HỌC AI HÔM NAY CHO DEV

1. **Tối ưu chi phí & hiệu năng LLM**
2. Để phát triển ứng dụng AI hiệu quả, các nhà phát triển cần biết cách tối ưu hóa chi phí và hiệu năng của mô hình ngôn ngữ lớn (LLM). Việc này giúp giảm thiểu chi phí tính toán và tăng tốc độ xử lý, từ đó cải thiện trải nghiệm người dùng. Các kỹ thuật như fine-tuning và LoRA có thể giúp đạt được mục tiêu này.
3. Ví dụ, bằng cách sử dụng kỹ thuật fine-tuning, chúng ta có thể tinh chỉnh mô hình LLM để phù hợp với nhiệm vụ cụ thể, giảm thiểu số lượng tham số cần thiết và tăng tốc độ xử lý.
4. Tip hoặc bước tiếp theo: Các nhà phát triển có thể bắt đầu bằng cách nghiên cứu các kỹ thuật như quantization và pruning để giảm thiểu kích thước mô hình và tăng tốc độ xử lý, từ đó tối ưu hóa hiệu năng và chi phí của ứng dụng AI.

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

Nguồn: Google News · Groq AI