BIS - MT · 22/05/2026

🤖 Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

✨ *"Take care of your body. It's the only place you have to live."*

↪ Hãy chăm sóc cơ thể bạn. Đó là nơi duy nhất bạn phải sống.

— Jim Rohn

💡 *Sức khỏe là nền tảng của mọi thành công — không có cơ thể và tinh thần khỏe mạnh, mọi tài sản và thành tích đều mất đi giá trị.*

📰 TIN TỨC NỔI BẬT

1

Lớp lệnh AI nhà kho đa tác nhân cho phép đạt được xuất sắc về hoạt động 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 giới thiệu Lớp lệnh AI Nhà kho Đa tác nhân, một công nghệ tiên tiến được thiết kế để tối ưu hóa hoạt động nhà kho và nâng cao trí tuệ chuỗi cung ứng. Giải pháp sáng tạo này sử dụng trí tuệ nhân tạo (AI) để tự động hóa quản lý nhà kho, cải thiện hiệu quả và giảm chi phí. Lớp lệnh AI Nhà kho Đa tác nhân là một trung tâm điều khiển tích hợp nhiều tác nhân AI để phân tích và tối ưu hóa các chức năng nhà kho khác nhau, bao gồm quản lý hàng tồn kho, thực hiện đơn hàng và hậu cần. Bằng cách tận dụng AI và các thuật toán học máy, hệ thống có thể dự đoán và ngăn chặn các nút thắt, tối ưu hóa phân bổ tài nguyên và cải thiện hiệu quả hoạt động tổng thể. Lợi ích chính của Lớp lệnh AI Nhà kho Đa tác nhân bao gồm khả năng hiển thị chuỗi cung ứng được cải thiện, độ chính xác hàng tồn kho được cải thiện và tốc độ thực hiện đơn hàng tăng. Công nghệ này cũng cho phép giám sát và phân tích hoạt động nhà kho theo thời gian thực, cho phép các doanh nghiệp đưa ra quyết định dựa trên dữ liệu và dẫn đầu trong cạnh tranh. Bằng cách tận dụng sức mạnh của AI, Lớp lệnh AI Nhà kho Đa tác nhân có khả năng cách mạng hóa cách thức hoạt động của các nhà kho và quản lý chuỗi cung ứng.

2 Nhà máy tác nhân: Kết nối các tác nhân, ứng dụng và dữ liệu với 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 sáng kiến mới nhằm kết nối các tác nhân, ứng dụng và dữ liệu thông qua các tiêu chuẩn mở. Dự án này sử dụng Nền tảng Điện toán Đám mây (MCP) và giao thức Ứng dụng-sang-Ứng dụng (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 tạo và tích hợp các tác nhân thông minh có thể giao tiếp với các ứng dụng và nguồn dữ liệu khác nhau. Sự tích hợp này cho phép tạo ra các hệ thống phức tạp và tự động hơn có thể thực hiện các nhiệm vụ phức tạp. Bằng cách tận dụng MCP và các tiêu chuẩn A2A, các nhà phát triển có thể xây dựng các giải pháp có thể mở rộng và bảo mật có thể được tích hợp dễ dàng với các hệ thống hiện có. Việc sử dụng các tiêu chuẩn mở trong Agent Factory thúc đẩy khả năng tương tác và linh hoạt, giúp các nhà phát triển dễ dàng xây dựng và triển khai các ứng dụng có thể tương tác với một loạt các tác nhân, ứng dụng và nguồn dữ liệu. Sáng kiến này có tiềm năng thúc đẩy đổi mới và tiến bộ trong các ngành công nghiệp khác nhau, từ chăm sóc sức khỏe đến tài chính, bằng cách cho phép tạo ra các hệ thống thông minh và kết nối hơn, sử dụng AI, API, LLM, model, token, và framework.

3 Giới thiệu Đặc tả Tác nhân Mở (Agent Spec): Một Biểu diễn Thống nhất cho các Tác nhân 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 tiêu chuẩn hóa cách thiết kế, phát triển và tích hợp các tác nhân AI vào các hệ thống khác nhau. Thông số kỹ thuật này cung cấp một khuôn khổ chung để mô tả hành vi, khả năng và tương tác của các tác nhân AI, cho phép giao tiếp và cộng tác liền mạch trên các nền tảng và ứng dụng khác nhau. Agent Spec được thiết kế để mở và có thể mở rộng, cho phép các nhà phát triển tạo ra các tác nhân tùy chỉnh có thể dễ dàng tích hợp với các hệ thống khác. Nó cung cấp một tập hợp các giao diện và

định dạng dữ liệu tiêu chuẩn hóa cho giao tiếp của tác nhân, giúp xây dựng và triển khai các ứng dụng được hỗ trợ bởi AI dễ dàng hơn. Bằng cách áp dụng Agent Spec, các nhà phát triển có thể tạo ra các hệ thống AI có khả năng tương tác và mở rộng hơn, giảm thiểu sự phức tạp và phân mảnh liên quan đến phát triển AI hiện tại. Agent Spec là một sáng kiến mã nguồn mở, và Oracle đã làm cho thông số kỹ thuật này có sẵn cho việc xem xét và phản hồi của công chúng. Mục tiêu là tạo ra một tiêu chuẩn được áp dụng rộng rãi cho biểu diễn tác nhân AI, thúc đẩy đổi mới và áp dụng trong hệ sinh thái AI.

4 Ngoài động cơ: 10 dự án mã nguồn mở định hình cách trò chơi thực sự được tạo ra

🇬🇧 *Beyond the engine: 10 open source projects shaping how games actually get made*

📰 GitHub Blog 🔗 Đọc bài viết →

GitHub, một nền tảng hàng đầu cho các nhà phát triển, đang nhấn mạnh tầm quan trọng của các dự án mã nguồn mở trong phát triển trò chơi. Mặc dù các công cụ game cung cấp nền tảng để tạo ra các trò chơi, nhưng có nhiều công cụ và quy trình làm việc tồn tại bên ngoài công cụ. Điều này bao gồm các đường ống tài sản, trình chỉnh sửa cấp độ, công cụ âm thanh và nhiều hơn nữa. Để giải quyết các điểm đau phổ biến trong phát triển trò chơi, GitHub đã xác định 10 dự án mã nguồn mở có thể giúp các nhà phát triển trò chơi với các nhiệm vụ như tạo nghệ thuật, hoạt hình, cấp độ, âm thanh và đối thoại. Những dự án này có thể được tích hợp vào bất kỳ công cụ trò chơi nào, bao gồm Godot, Unity, Unreal, MonoGame và các công cụ tùy chỉnh. 10 dự án mã nguồn mở được đề cập bao gồm Blockbench, một trình chỉnh sửa mô hình 3D cho các mô hình poly thấp với kết cấu nghệ thuật pixel, và các công cụ khác cho gỡ lỗi, tạo UI, và quản lý tài sản. Những dự án này được tạo ra để giải quyết các điểm đau cụ thể trong phát triển trò chơi và có thể được sử dụng để cải thiện năng suất và hợp tác giữa các nhà phát triển. Bằng cách tận dụng các công cụ mã nguồn mở, các nhà phát triển trò chơi có thể tối ưu hóa quy trình làm việc của mình và tập trung vào việc tạo ra các trò chơi chất lượng cao. Việc sử dụng các dự án mã nguồn mở cũng có thể giúp giải quyết các thách thức trong DevSecOps và cải thiện trải nghiệm tổng thể của nhà phát triển.

5

AiFinPay: Thanh toán tự động cho ruvnet/ruflo

🇬🇧 *AiFinPay: Autonomous Payments for ruvnet/ruflo*

📰 Dev.to AI 🔗 Đọc bài viết →

> AiFinPay, một SDK thanh toán một dòng, đã được tích hợp với ruvnet/ruflo, một nền tảng điều phối đại lý, để cho phép thanh toán không gián đoạn cho các đại lý AI. Sự hợp tác này nhằm mục đích cách mạng hóa cách các đại lý AI tương tác và giao dịch, mở ra những khả năng mới cho các quy trình tự động và hệ thống AI đối thoại. Sự tích hợp của AiFinPay cho phép thanh toán đại lý AI được tối ưu hóa, quyền cho tương lai của sự đổi mới được thúc đẩy bởi AI. Sự hợp tác giữa AiFinPay và ruvnet/ruflo dự kiến sẽ mang lại những tiến bộ đáng kể trong lĩnh vực AI và thanh toán tự động.

6

Giới thiệu Gia đình Tác nhân Ettin Reranker

🇬🇧 *Introducing the Ettin Reranker Family*

📰 Hugging Face Blog 🔗 Đọc bài viết →

> Các nhà nghiên cứu đã giới thiệu gia đình Ettin Reranker, một tập hợp gồm sáu mô hình Sentence Transformers CrossEncoder reranker hiện đại. Những mô hình này được xây dựng trên nền tảng các bộ mã hóa ModernBERT của Ettin và được thiết kế để cải thiện độ chính xác của các nhiệm vụ xếp hạng văn bản. Các mô hình reranker này được đào tạo bằng cách sử dụng công thức chưng cất và có thể được kết hợp với các mô hình embedder khác để đạt được hiệu suất tối ưu. Các mô hình này có sẵn để tải xuống, cùng với dữ liệu và công thức đào tạo được sử dụng để tạo ra chúng. Gia đình Ettin Reranker bao gồm sáu mô hình với các kích cỡ khác nhau, mỗi mô hình có điểm mạnh và điểm yếu riêng. Các mô hình này có thể được sử dụng cho một loạt các nhiệm vụ, bao gồm xếp hạng văn bản và thu hồi. Chúng được thiết kế để sử dụng kết hợp với các mô hình khác, chẳng hạn như mô hình nhúng, để cải thiện độ chính xác tổng thể của hệ thống. Các nhà nghiên cứu cũng đã cung cấp một hướng dẫn chi tiết về cách sử dụng các mô hình, bao gồm hướng dẫn về cách cài đặt và đào tạo chúng. Các mô hình này tương thích với nhiều framework khác nhau và có thể được sử dụng với chỉ một vài dòng mã. Các nhà nghiên cứu cũng đã cung cấp một ví dụ hoàn chỉnh về cách sử dụng các mô hình trong một ứng dụng thực tế.

7

Tác nhân Datasette

🇬🇧 *Datasette Agent*

📰 Simon Willison 🔗 Đọc bài viết →

> Đại lý Datasette, một trợ lý AI mới cho Datasette, đã được phát hành. Interface trò chuyện này cho phép người dùng đặt câu hỏi về dữ liệu của họ được lưu trữ trong Datasette, với tùy chọn tạo biểu đồ bằng plugin datasette-agent-charts. Thông báo bao gồm một video demo giới thiệu khả năng của đại lý chống lại các cơ sở dữ liệu ví dụ. Datasette Agent được xây dựng trên một thư viện Python cho các mô hình ngôn ngữ lớn (LLM) và có thể mở rộng bằng plugin. Tác giả đã sử dụng thành công Claude Code và OpenAI Codex để viết plugin và đang khám phá việc sử dụng các mô hình cục bộ. Việc phát hành Datasette Agent cũng đã thông báo cho việc phát triển refactor LLM 0.32a0 chính và đã truyền cảm hứng cho việc tạo ra một trợ lý AI cá nhân, Claw. Dự án này mở cho phản hồi và hợp tác, với kênh Discord có sẵn để thảo luận. Datasette Agent sẽ được triển khai cho người dùng Datasette Cloud trong tương lai.

8

Các cuộc thảo luận tròn: Liệu AI có thể học để hiểu Thế giới?

🇬🇧 *Roundtables: Can AI Learn to Understand the World?*

📰 MIT Tech Review 🔗 Đọc bài viết →

> Một cuộc thảo luận gần đây giữa các chuyên gia AI tại MIT Technology Review đã khám phá khả năng trí tuệ nhân tạo (AI) hiểu thế giới vật lý. Cuộc trò chuyện, do Biên tập viên trưởng Mat Honan, Biên tập viên cao cấp Will Douglas Heaven và Báo cáo viên AI Grace Huckins dẫn dắt, đã đi sâu vào những hạn chế của các Large Language Models (LLMs) hiện tại và tiềm năng của các model thế giới để vượt qua những hạn chế này. Cuộc thảo luận đã nhấn mạnh sự phát triển nhanh chóng của AI, như được lưu ý trong Chỉ số AI năm 2026 của Stanford, cho thấy AI đang tiến bộ với tốc độ chưa từng có. Cuộc trò chuyện cũng đề cập đến những ý nghĩa của những phát triển này và nhu cầu của các hệ thống AI phải hiểu tốt hơn thế giới bên ngoài.

⚡ TIPS & TRICKS CHO DEV

⚡ Kết nối File System

Vấn đề: Cần truy cập file cục bộ trong ứng dụng AI.

Cách làm: Sử dụng MCP để kết nối với file system, cho phép đọc và ghi file. Ví dụ: `Claude Desktop` có thể truy cập file hệ thống thông qua MCP.
Đánh giá: Hiệu quả cao, nên dùng khi cần tương tác với file cục bộ.

⚡ Tích hợp Git

Vấn đề: Cần tự động hóa quá trình commit và push code.
Cách làm: Sử dụng MCP để kết nối với Git, cho phép tự động hóa commit và push code. Ví dụ: `Claude Code` có thể commit và push code thông qua MCP.
Đánh giá: Tiết kiệm thời gian, nên dùng khi cần tự động hóa quản lý code.

⚡ Truy cập Database

Vấn đề: Cần truy cập và quản lý dữ liệu trong database.
Cách làm: Sử dụng MCP để kết nối với database, cho phép đọc và ghi dữ liệu. Ví dụ: `Cursor` có thể truy cập database thông qua MCP.
Đánh giá: Hiệu quả cao, nên dùng khi cần tương tác với dữ liệu trong database.

📚 **BÀI HỌC AI HÔM NAY CHO DEV**

1. **Tối ưu chi phí & hiệu năng LLM**
2. Để xây dựng ứng dụng AI hiệu quả, các developer 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 dữ liệu. Điều này đặc biệt quan trọng khi triển khai ứng dụng trên các thiết bị di động hoặc trên các môi trường có tài nguyên hạn chế.
3. Ví dụ, việc 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. Ngoài ra, việc sử dụng các mô hình LLM cục bộ (local LLM) hoặc trí tuệ nhân tạo trên biên (edge AI) cũng có thể giúp giảm thiểu chi phí và tăng tốc độ xử lý.
4. 💡 Tip hoặc bước tiếp theo: Hãy thử nghiệm với các kỹ thuật tối ưu hóa khác nhau để tìm ra giải pháp phù hợp cho ứng dụng của bạn, chẳng hạn như sử dụng các mô hình LLM nhỏ hơn hoặc tối ưu hóa mã nguồn để giảm thiểu chi phí tính toán.