BIS - MT · 16/05/2026

🤖 Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

✨ *"Talent wins games, but teamwork and intelligence wins championships."*

↪ Tài năng giúp thắng trận, nhưng tinh thần đồng đội và trí tuệ mới giành được chức vô địch.

— Michael Jordan

💡 *Cá nhân xuất sắc tạo ra kết quả tốt, nhưng đội nhóm gắn kết và phối hợp tốt mới tạo ra kết quả vĩ đại và bền vững.*

📰 TIN TỨC NỔI BẬT

1

Xây dựng agent phân tích tài chính thông minh với LangGraph và Strands Agents | Amazon Web Services

🇬🇧 *Build an intelligent financial analysis agent with LangGraph and Strands Agents | Amazon Web Services*

📰 Amazon Web Services (AWS) 🔗 Đọc bài viết →

> Amazon Web Services (AWS) đã giới thiệu một giải pháp mới để xây dựng một tác nhân phân tích tài chính thông minh sử dụng LangGraph và Strands Agents. Phương pháp đổi mới này cho phép người dùng tạo một công cụ phân tích tài chính tinh vi có thể xử lý và phân tích lượng lớn dữ liệu tài chính. LangGraph là một thư viện xử lý ngôn ngữ tự nhiên (NLP) cho phép người dùng trích xuất thông tin chi tiết từ các báo cáo và tuyên bố tài chính. Nó có thể xác định thông tin chính như doanh thu, chi phí và lợi nhuận, và cung cấp một biểu diễn cấu trúc của dữ liệu. Strands Agents, mặt khác, là một nền tảng low-code cho phép người dùng tạo các ứng dụng và quy trình tùy chỉnh. Bằng cách kết hợp LangGraph với Strands Agents, người dùng có thể tạo một tác nhân phân tích tài chính toàn diện có thể cung cấp thông tin chi tiết và khuyến nghị theo thời gian thực. Giải pháp này được thiết kế để giúp các doanh nghiệp và tổ chức đưa ra quyết định dựa trên dữ liệu bằng cách cung cấp sự hiểu biết sâu sắc hơn về hiệu suất tài chính của họ. Tác nhân phân tích tài chính thông minh có thể được tích hợp với các hệ thống và công cụ tài chính khác nhau, khiến nó trở thành một tài sản quý giá cho phân tích và lập kế hoạch tài chính.

2 Bên trong Agent2Agent (A2A) Protocol của Google: Dạy các AI agent giao tiếp với nhau

🇬🇧 *Inside Google's Agent2Agent (A2A) Protocol: Teaching AI Agents to Talk to Each Other*

📰 towardsdatascience.com 🔗 Đọc bài viết →

> Google đã phát triển một giao thức nội bộ gọi là Agent2Agent (A2A), cho phép các tác nhân trí tuệ nhân tạo (AI) giao tiếp với nhau một cách liền mạch. Giao thức A2A là một thành phần quan trọng trong hệ sinh thái AI của Google, cho phép các tác nhân AI khác nhau trao đổi thông tin và phối hợp hành động của chúng. Giao thức A2A được thiết kế để tạo điều kiện cho giao tiếp hiệu quả và hiệu suất giữa các tác nhân AI, ngay cả khi chúng được phát triển bằng các ngôn ngữ lập trình hoặc framework khác nhau. Điều này cho phép các tác nhân AI của Google làm việc cùng nhau trên các nhiệm vụ phức tạp, chẳng hạn như xử lý ngôn ngữ tự nhiên và thị giác máy tính. Bằng cách dạy các tác nhân AI nói chuyện với nhau, Google nhằm tạo ra một hệ sinh thái AI gắn kết và hợp tác hơn. Giao thức A2A có tiềm năng cách mạng hóa cách các tác nhân AI tương tác với nhau, cho phép chúng giải quyết các vấn đề phức tạp và cải thiện hiệu suất tổng thể. Giao thức A2A là công nghệ nội bộ của Google, và như vậy, các hoạt động và ứng dụng chính xác của nó không được công bố công khai. Tuy nhiên, sự phát triển của nó nhấn mạnh nỗ lực liên tục của Google trong việc thúc đẩy lĩnh vực trí tuệ nhân tạo và tạo ra các hệ thống AI tinh vi hơn.

3 Tăng tốc phát triển với Amazon Bedrock AgentCore MCP server | Trí tuệ nhân tạo

🇬🇧 *Accelerate development with the Amazon Bedrock AgentCore MCP server | Artificial Intelligence*

📰 Amazon Web Services (AWS) 🔗 Đọc bài viết →

> Amazon Web Services (AWS) đã giới thiệu máy chủ Amazon Bedrock AgentCore MCP, được thiết kế để tăng tốc phát triển trong lĩnh vực trí tuệ nhân tạo (AI). Máy chủ này là một thành phần chính của nền tảng Amazon Bedrock, nhằm mục đích đơn giản hóa quá trình xây dựng, triển khai và quản lý các model AI. Máy chủ Amazon Bedrock AgentCore MCP cung cấp một môi trường phát triển AI có khả năng mở rộng và bảo mật, cho phép các nhà phát triển tập trung vào việc xây dựng và đào tạo các model mà không phải lo lắng về cơ sở hạ tầng cơ bản. Với máy chủ này, các nhà phát triển có thể dễ dàng triển khai và quản lý các model AI, cũng như tích hợp chúng với các dịch vụ AWS khác. Máy chủ AgentCore MCP được xây dựng trên kiến trúc

microservices, cho phép nó xử lý các khối lượng công việc AI phức tạp và cung cấp khả năng tính toán hiệu suất cao. Điều này cho phép các nhà phát triển nhanh chóng lặp lại và tinh chỉnh các model AI của họ, tăng tốc quá trình phát triển và giảm thời gian đưa sản phẩm ra thị trường. Bằng cách tận dụng máy chủ Amazon Bedrock AgentCore MCP, các nhà phát triển có thể tối ưu hóa quy trình phát triển AI của họ, cải thiện hợp tác và tăng năng suất, cuối cùng dẫn đến sự đổi mới nhanh hơn và kết quả kinh doanh tốt hơn.

4

Trải nghiệm tài chính cá nhân mới trên ChatGPT

🇬🇧 *A new personal finance experience in ChatGPT*

📰 OpenAI Blog 🔗 Đọc bài viết →

OpenAI đang phát hành bản xem trước của một trải nghiệm tài chính cá nhân mới trong ChatGPT dành cho người dùng Pro tại Mỹ. Tính năng này cho phép người dùng kết nối tài khoản tài chính của mình một cách bảo mật, xem bảng điều khiển chi tiêu của họ và đặt câu hỏi cho ChatGPT dựa trên ngữ cảnh tài chính của họ. Mục tiêu là cung cấp sự hiểu biết toàn diện hơn về tình hình tài chính của mình, giúp người dùng phát hiện ra các mẫu, hiểu sự đánh đổi và lên kế hoạch cho các quyết định quan trọng. Với khả năng kết nối hơn 12.000 tổ chức tài chính, người dùng có thể liên kết tài khoản của mình thông qua hỗ trợ Plaid và hỗ trợ Intuit sẽ có sẵn sớm. Khi đã kết nối, người dùng có thể chia sẻ ngữ cảnh quan trọng về cuộc sống tài chính của mình, chẳng hạn như mục tiêu tiết kiệm hoặc mua hàng sắp tới, để thông báo cho các cuộc trò chuyện trong tương lai. Trải nghiệm này được thiết kế để giúp người dùng quản lý tài chính của mình hiệu quả hơn, nhưng nó không thay thế cho lời khuyên tài chính chuyên nghiệp. Tính năng này hiện có sẵn cho người dùng Pro tại Mỹ và sẽ được mở rộng một cách cẩn thận sau khi học hỏi từ việc sử dụng trong thế giới thực.

5

Musk đấu Altman tuần 3: Hai bên công kích uy tín của nhau. Giờ đây bồi thẩm đoàn sẽ chọn phe.

🇬🇧 *Musk v. Altman week 3: Musk and Altman traded blows over each other's credibility. Now the jury will pick a side.*

📰 MIT Tech Review 🔗 Đọc bài viết →

Trong tuần thứ ba của vụ xét xử nổi tiếng giữa Elon Musk và CEO OpenAI Sam Altman, các luật sư của cả hai bên đã trình bày lập luận

cuối cùng và xung đột về tính hợp pháp của nhau. Luật sư của Musk, Steven Molo, đã buộc tội Altman nói dối và tư lợi, dẫn chứng lịch sử bị cáo buộc không trung thực và đầu tư cá nhân vào các công ty khởi nghiệp làm ăn với OpenAI. Altman đã phản pháo, mô tả Musk như một người tìm kiếm quyền lực muốn kiểm soát sự phát triển của trí tuệ nhân tạo tổng quát (AGI). Vụ xét xử tập trung vào việc tái cấu trúc OpenAI năm 2025, chuyển đổi một công ty con có lợi nhuận thành một công ty lợi ích công cộng, và tuyên bố của Musk rằng Altman và chủ tịch OpenAI Greg Brockman đã phá vỡ lời hứa giữ OpenAI như một tổ chức phi lợi nhuận. Luật sư của Altman, Sarah Eddy, đã lập luận rằng Musk đã kiện quá muộn và động cơ thực sự của anh ta là phá hoại một đối thủ cạnh tranh với công ty AI của riêng mình, xAI. Hội đồng giám khảo sẽ bắt đầu thảo luận vào thứ Hai và đưa ra phán quyết tư vấn sớm nhất vào tuần tới, điều này sẽ không ràng buộc đối với thẩm phán. Kết quả của vụ xét xử có thể có những ý nghĩa quan trọng đối với tương lai của OpenAI và tiềm năng IPO với định giá gần 1 nghìn tỷ đô la.

6

Tự động hóa với tốc độ của Swamp

🇬🇧 *Automation at the speed of Swamp*

📰 Changelog 🔗 Đọc bài viết →

Trong một tập gần đây, người sáng lập công nghệ Adam Jacob đã thảo luận về tác động của các tác nhân AI đối với phát triển phần mềm. Ông đã chia sẻ kinh nghiệm của mình với Swamp, một công cụ cho phép nhóm 18 người của ông giảm xuống còn năm thành viên trong khi vẫn giao 900 bản cập nhật chỉ trong bốn tuần. Jacob nhấn mạnh tầm quan trọng của kiến trúc phần mềm và thiết kế hướng lĩnh vực, cho rằng những điều này hiện nay quan trọng hơn kỹ năng lập trình. Ông cũng đã chứng minh khả năng của Swamp bằng cách sử dụng nó để tự động hóa các nhiệm vụ trên hộp Proxmox của mình. Ngoài ra, Jacob đã giải thích quyết định của mình trong việc đưa lại Kiểm tra Chấp nhận Người dùng (UAT) và tiết lộ rằng ông không bao giờ chấp nhận yêu cầu kéo (pull request) đối với Swamp. Tập này cũng có các cuộc thảo luận về các công cụ và nền tảng khác nhau, bao gồm Coder, Tailscale, RWX và Fly.io.

7

Granite Embedding Multilingual R2: Multilingual Embeddings mã nguồn mở Apache 2.0 với 32K Context — Chất lượng Retrieval tốt nhất trong phân khúc dưới 100M

🇬🇧 *Granite Embedding Multilingual R2: Open Apache 2.0 Multilingual Embeddings with 32K Context — Best Sub-100M Retrieval Quality*

📰 Hugging Face Blog 🔗 Đọc bài viết →

> Granite Embedding Multilingual R2 là một bản phát hành quan trọng trong lĩnh vực các model nhúng đa ngôn ngữ. Hai model mới, một model compact 97M-parameter và một model đầy đủ 311M, đã được phát triển để giải quyết sự căng thẳng dai dẳng giữa phạm vi ngôn ngữ rộng và kích thước model. Các model này hỗ trợ 200+ ngôn ngữ, với chất lượng thu hồi được cải thiện cho 52 ngôn ngữ và mã lập trình. Chúng có thể xử lý độ dài ngữ cảnh lên đến 32.768 token, tăng 64 lần so với người tiền nhiệm. Các model này được phát hành theo giấy phép Apache 2.0 và tương thích với các framework phổ biến như LangChain, LlamaIndex, Haystack và Milvus. Chúng cũng đi kèm với trọng số ONNX và OpenVINO cho suy luận CPU-optimized. Model compact vượt trội so với mọi trình nhúng đa ngôn ngữ mở dưới 100M trên MTEB Multilingual Retrieval, trong khi model đầy đủ đạt điểm 65,2 trên MTEB Multilingual Retrieval, xếp thứ hai trong số các model mở dưới 500M tham số.

8

Một lần AI code review là chưa đủ. Đây là vòng lặp thực sự giúp bắt lỗi.

🇬🇧 *One AI code review pass isn't enough. Here's the loop that actually catches bugs.*

📰 Dev.to AI 🔗 Đọc bài viết →

> Các công cụ xem xét mã code AI, chẳng hạn như Claude và Copilot, thường cung cấp một phản hồi sạch sẽ trong lần đầu tiên, khiến các developer tự tin khi hợp nhất mã code. Tuy nhiên, nghiên cứu cho thấy một lần xem xét của AI là kém hơn về mặt thống kê so với lần đầu tiên của một con người mệt mỏi. Điều này không phải do sự thông minh của model, mà là cách thức hoạt động của việc xem xét. Các model AI quét tìm các vấn đề rõ ràng, chẳng hạn như thụt lề sai hoặc biến không sử dụng, nhưng thường bỏ qua các lỗi phức tạp hơn có thể gây tốn kém đáng kể. Để cải thiện việc xem xét mã code AI, các developer nên sử dụng một vòng lặp gồm năm yêu cầu, mỗi yêu cầu xem xét cùng một đoạn mã code với một câu hỏi khác nhau. Cách tiếp cận này buộc model phải tích cực tìm kiếm các vấn đề hoặc rõ ràng tuyên bố rằng không có vấn đề nào. Bằng cách làm như vậy, model được ngăn chặn khỏi việc đồng ý mặc định khi không tìm thấy vấn đề rõ ràng.

> Vòng lặp này có thể được thực hiện trong môi trường Tích hợp Liên tục (CI), với chi phí khoảng 0,10 đô la cho một yêu cầu kéo 200 dòng. Bằng cách sử dụng cách tiếp cận này, các developer có thể bắt được nhiều lỗi hơn và cải thiện chất lượng mã code của họ.

⚡ TIPS & TRICKS CHO DEV

⚡ Tự động sinh test case

Vấn đề: Thiếu test case chất lượng để kiểm tra mã nguồn.
Cách làm: Sử dụng AI để sinh test case tự động với lệnh `python -m unittest discover`, hoặc prompt như "Tạo test case cho hàm tính tổng".
Đánh giá: Hiệu quả với mã nguồn lớn, tiết kiệm thời gian.

⚡ Code review tự động

Vấn đề: Code review thủ công tốn thời gian và dễ bỏ sót lỗi.
Cách làm: Sử dụng công cụ như GitHub Code Review với lệnh `git push origin main`, hoặc prompt "Đánh giá mã nguồn này".
Đánh giá: Tăng tốc độ và chất lượng code review, nhưng cần cấu hình phù hợp.

⚡ QA automation

Vấn đề: Kiểm tra chất lượng sản phẩm thủ công không hiệu quả.
Cách làm: Sử dụng AI để tự động hóa kiểm tra với công cụ như Selenium, lệnh `java -jar selenium.jar`.
Đánh giá: Tiết kiệm thời gian và tăng hiệu quả kiểm tra, nhưng cần đầu tư ban đầu.

📚 BÀI HỌC AI HÔM NAY CHO DEV

1. **Tích hợp AI API vào ứng dụng**
2. Tích hợp AI API vào ứng dụng giúp các nhà phát triển tăng cường khả năng xử lý và phân tích dữ liệu, cũng như cung cấp trải nghiệm người dùng thông minh hơn. Việc này cho phép ứng dụng của họ học hỏi và thích nghi với nhu cầu người dùng. Các nhà phát triển cần biết cách tích hợp AI API để tận dụng lợi thế của công nghệ này.
3. Ví dụ, một ứng dụng có thể sử dụng API của Google Cloud Vision để nhận diện và phân loại hình ảnh, hoặc sử dụng API của Dialogflow để tạo ra chatbot thông minh.
4. 💡 Tip: Để bắt đầu tích hợp AI API, hãy lựa chọn một nền tảng phù hợp với nhu cầu của ứng dụng và bắt đầu với các ví dụ code đơn giản, sau đó dần dần mở rộng tính năng và phức tạp của ứng dụng.

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

Nguồn: Google News · Groq AI