BIS - MT · 25/05/2026

🤖 Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

✨ *"Your present circumstances don't determine where you can go; they merely determine where you start."*

↪ Hoàn cảnh hiện tại không quyết định bạn có thể đi đến đâu; nó chỉ xác định điểm xuất phát của bạn.

— Nido Qubein

💡 *Xuất phát điểm không quan trọng bằng hướng đi và nỗ lực — nhiều người thành công đã bắt đầu từ hoàn cảnh rất khó khăn.*

📰 TIN TỨC NỔI BẬT

1

Bên trong giao thức Agent2Agent (A2A) của Google: Dạy các agent AI trò chuyện với nhau

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

📰 Towards Data Science 🔗 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 được thiết kế để tạo điều kiện cho cuộc trò chuyện giữa các tác nhân AI, cho phép họ chia sẻ thông tin và phối hợp hành động hiệu quả hơn. Giao thức này được xây dựng trên một đồ thị tri thức, đóng vai trò là một khuôn khổ hiểu biết chung cho các tác nhân AI. Cơ sở tri thức chung này cho phép các tác nhân hiểu được ngữ cảnh và ý định của nhau, làm cho các tương tác của họ trở nên hiệu quả và chính xác hơn. A2A là một thành phần quan trọng trong các nỗ lực nghiên cứu AI rộng lớn hơn của Google, đặc biệt là trong các lĩnh vực như xử lý ngôn ngữ tự nhiên và hệ thống đa tác nhân. Bằng cách dạy các tác nhân AI giao tiếp với nhau, Google nhằm tạo ra các hệ thống tinh vi và tự chủ hơn có thể giải quyết các nhiệm vụ và vấn đề phức tạp. Giao thức A2A có ý nghĩa quan trọng đối với nhiều ứng dụng, bao gồm cả rô-bốt trò chuyện dịch vụ khách hàng, trợ lý ảo và phương tiện tự hành. Bằng cách cho phép các tác nhân AI làm việc cùng nhau một cách liền mạch, Google đang đẩy ranh giới của nghiên cứu và phát triển AI.

2 Bộ công cụ phát triển agent: Làm cho việc xây dựng ứng dụng đa agent trở nên dễ dàng

Agent Development Kit: Making it easy to build multi-agent applications

blog.google Đọc bài viết →

Google đã giới thiệu Bộ công cụ phát triển Trợ lý (ADK), một công cụ được thiết kế để đơn giản hóa việc tạo ra các ứng dụng đa trợ lý. ADK cho phép các nhà phát triển xây dựng và quản lý nhiều trợ lý, là các chương trình phần mềm có thể thực hiện các nhiệm vụ cụ thể và tương tác với nhau. Với ADK, các nhà phát triển có thể tạo ra các trợ lý có thể được sử dụng trên nhiều nền tảng khác nhau, bao gồm Trợ lý Google và các dịch vụ của bên thứ ba khác. Bộ công cụ này cung cấp một tập hợp các API và công cụ cho phép các nhà phát triển xây dựng, thử nghiệm và triển khai các trợ lý một cách nhanh chóng và hiệu quả. ADK cũng hỗ trợ việc sử dụng khả năng xử lý ngôn ngữ tự nhiên (NLP) và học máy (ML), cho phép các trợ lý hiểu và phản hồi đầu vào của người dùng một cách trực quan và cá nhân hóa hơn. Bằng cách cung cấp một khuôn khổ tiêu chuẩn cho việc xây dựng các ứng dụng đa trợ lý, ADK nhằm mục đích giúp các nhà phát triển tạo ra những trải nghiệm phức tạp và tinh vi hơn có thể được sử dụng trong nhiều ngữ cảnh khác nhau, bao gồm cả việc tích hợp với AI, API, LLM và các model khác, cũng như token và framework để hỗ trợ cho quá trình phát triển.

3 Tăng tốc phát triển với máy chủ AgentCore MCP của Amazon Bedrock | 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 AgentCore MCP là một thành phần chính của nền tảng Amazon Bedrock, cho phép các tổ chức xây dựng, đào tạo và triển khai các model AI với quy mô lớn. Máy chủ AgentCore MCP là một máy chủ hiệu suất cao cung cấp môi trường phát triển AI có khả năng mở rộng và bảo mật. Nó được tối ưu hóa cho các công việc Machine Learning (ML) và hỗ trợ nhiều framework và công cụ AI khác nhau. Với máy chủ AgentCore MCP, các nhà phát triển có thể tăng tốc phát triển model AI, giảm thời gian đào tạo và cải thiện độ chính xác của model. Máy chủ Amazon Bedrock AgentCore MCP được

tích hợp với các dịch vụ AWS khác, chẳng hạn như SageMaker và Lake Formation, để cung cấp một nền tảng phát triển AI toàn diện. Sự tích hợp này cho phép các nhà phát triển truy cập vào nhiều công cụ và dịch vụ AI khác nhau, giúp dễ dàng xây dựng và triển khai các model AI. Bằng cách tận dụng máy chủ Amazon Bedrock AgentCore MCP, các tổ chức có thể tăng tốc phát triển và triển khai AI, thúc đẩy đổi mới và tăng trưởng kinh doanh.

4 Một nhóm tin tặc đang đầu độc mã nguồn mở với quy mô chưa từng có

🇬🇧 *A hacker group is poisoning open source code at an unprecedented scale*

📰 Ars Technica 🔗 Đọc bài viết →

Một nhóm hacker nổi tiếng có tên TeamPCP đã thực hiện một loạt các cuộc tấn công chuỗi cung ứng phần mềm với quy mô chưa từng có. Nhóm này đã làm hỏng hàng trăm công cụ mã nguồn mở, tống tiền các nạn nhân để kiếm lợi và gieo rắc sự mất lòng tin trong hệ sinh thái phần mềm. Nạn nhân mới nhất là GitHub, một nền tảng thuộc sở hữu của Microsoft, nơi các hacker đã truy cập vào khoảng 4.000 kho mã sau khi một nhà phát triển cài đặt một tiện ích mở rộng "độc hại" cho trình soạn thảo mã VSCode. TeamPCP đã tuyên bố đã truy cập và quảng cáo mã nguồn và tổ chức nội bộ của GitHub để bán trên một diễn đàn tội phạm mạng. Theo công ty an ninh mạng Socket, TeamPCP đã thực hiện 20 "làn sóng" tấn công chuỗi cung ứng trong vài tháng qua, che giấu malware trong hơn 500 phần mềm riêng biệt. Điều này đã cho phép nhóm này xâm phạm hàng trăm công ty và trở thành cuộc tấn công chuỗi cung ứng phần mềm dài nhất từ trước đến nay. Các chiến thuật của nhóm này bao gồm việc truy cập vào một mạng nơi một công cụ mã nguồn mở đang được phát triển, cài đặt malware và đánh cắp thông tin đăng nhập để xuất bản các phiên bản độc hại của các công cụ. TeamPCP cũng đã tự động hóa các cuộc tấn công của mình bằng một loại giun tự lan truyền có tên là Mini Shai-Hulud.

5 GitHub được công nhận là Nhà lãnh đạo trong Quadrant Magic của Gartner về các agent mã hóa AI doanh nghiệp lần thứ ba liên tiếp

🇬🇧 *GitHub recognized as a Leader in the Gartner® Magic Quadrant™ for Enterprise AI Coding Agents for the third year in a row*

📰 GitHub Blog 🔗 Đọc bài viết →

GitHub đã được công nhận là Nhà lãnh đạo trong Gartner Magic Quadrant về Các tác nhân mã hóa AI doanh nghiệp lần thứ ba liên tiếp. Thành tựu này nhấn mạnh cam kết của GitHub trong việc trao quyền cho các developer với một nền tảng mở, bảo mật và được hỗ trợ bởi AI, định hình lại tương lai của phát triển phần mềm. Công cụ GitHub Copilot của công ty cho phép các developer giao nhiệm vụ cho một tác nhân và rời đi, cho phép tác nhân xử lý phần còn lại, dẫn đến việc phát triển phần mềm nhanh hơn. Theo Gartner, các quy trình làm việc của tác nhân mã hóa AI không đồng bộ sẽ cải thiện năng suất của các đội kỹ sư phần mềm từ 30% đến 50% vào năm 2028. GitHub tin rằng việc đạt được những lợi ích này đòi hỏi phải có khả năng tác nhân trên mọi giai đoạn của chu kỳ phát triển phần mềm (SDLC), không chỉ là tạo mã. Sự tập trung của công ty vào việc tạo mã AI, xem xét, bảo mật và quản lý nhằm giải quyết các thách thức cốt lõi trong DevSecOps và cho phép các developer xây dựng, vận hành và duy trì phần mềm một cách hiệu quả hơn. Việc công nhận GitHub là Nhà lãnh đạo trong Gartner Magic Quadrant nhấn mạnh vị trí của nó như một nền tảng developer hàng đầu, cung cấp trải nghiệm hiệu suất cao và luôn sẵn sàng trên toàn hệ sinh thái. Công ty tiếp tục đổi mới và mở rộng khả năng của mình, cung cấp cho các developer các công cụ và tài nguyên cần thiết để phát triển kỹ năng và sự nghiệp của họ.

6 Các nhà phát triển: Hãy xác thực ứng dụng di động y tế của bạn với IEEE

🇬🇧 *Developers: Get Your Medical Mobile App Verified By IEEE*

📰 IEEE Spectrum 🔗 Đọc bài viết →

Các ứng dụng di động y tế, được sử dụng để quản lý các tình trạng như trầm cảm và đau mãn tính, thường thiếu sự xác minh từ các cơ quan quản lý. Trong số hơn 55.000 ứng dụng y tế, hầu hết chưa được đánh giá về tính vững chắc kỹ thuật, thiết kế đạo đức hoặc lợi ích lâm sàng. Các ứng dụng này thường không tuân thủ các luật bảo mật và quyền riêng tư của dữ liệu, đặt thông tin sức khỏe nhạy cảm của người dùng vào tình trạng rủi ro. Hiệp hội Tiêu chuẩn IEEE đã ra mắt Sổ đăng ký và Đánh giá Ứng dụng Di động Y tế Toàn cầu IEEE để giải quyết vấn đề này. Thư mục tìm kiếm công khai liệt kê các ứng dụng đã được các chuyên gia thẩm định theo nhiều tiêu chí, bao gồm tính vững chắc kỹ thuật, thiết kế đạo đức và hiệu quả lâm sàng. Mục tiêu là thiết lập một phương pháp đánh giá tiêu chuẩn hóa sử dụng các tiêu chí được phát triển bởi các chuyên gia, giúp bệnh nhân, nhà lâm sàng và

hệ thống chăm sóc sức khỏe phân biệt ứng dụng liệu pháp đáng tin cậy với ứng dụng được tiếp thị tốt. Sổ đăng ký IEEE nhằm lấp đầy khoảng trống trong việc giám sát quản lý và cung cấp một nguồn tin cậy để xác minh các ứng dụng di động y tế.

7 Dừng việc nhập lệnh. Bắt đầu chỉ định: Cách phát triển dựa trên spec sửa lỗi mã hóa AI

🇬🇧 *Stop Prompting. Start Specifying: How Spec-Driven Development Fixes AI Coding*

📰 Dev.to AI 🔗 Đọc bài viết →

Tình trạng hiện tại của việc mã hóa được hỗ trợ bởi AI đang bị ảnh hưởng bởi "khủng hoảng ngữ cảnh AI", nơi các công cụ AI quên các quyết định trước đó, bỏ qua các mẫu đã được thiết lập và tái tạo hành vi im lặng do ngữ cảnh tạm thời và ý định không rõ ràng. Điều này dẫn đến các cuộc trò chuyện bị phân mảnh, kiến trúc trôi dạt và tạo ra các tính năng không mong muốn. Để giải quyết vấn đề này, một framework mới gọi là OpenSpec đã được giới thiệu, cho phép phát triển theo hướng spec (SDD). OpenSpec là một framework nhẹ, dựa trên tệp, chứa các thông số kỹ thuật, thiết kế và nhiệm vụ có cấu trúc cho các tính năng, thí nghiệm và tái cấu trúc. Nó cung cấp một hệ thống kiến thức nhất quán và các yêu cầu rõ ràng mà tồn tại qua các phiên và mở rộng quy mô với dự án. OpenSpec hoạt động với hơn 20 công cụ mã hóa AI và không bị khóa vào một hệ sinh thái. Bằng cách sử dụng OpenSpec, các nhà phát triển có thể định nghĩa các thông số kỹ thuật rõ ràng trở thành nguồn thông tin duy nhất cho các đề xuất, đánh giá và tự động hóa, biến AI từ một công cụ đoán thành một người xây dựng hệ thống đáng tin cậy.

8 Tải xuống: Tương lai của mã hóa, 'Vận động viên Olympic steroid' và khoa học được thúc đẩy bởi AI

🇬🇧 *The Download: coding's future, the 'Steroid Olympics,' and AI-driven science*

📰 MIT Tech Review 🔗 Đọc bài viết →

Trong phiên bản mới nhất của The Download, một bản tin công nghệ hàng ngày, một số câu chuyện chính được nhấn mạnh. Tại sự kiện của nhà phát triển Anthropic, nó đã được tiết lộ rằng gần một nửa số người tham dự đã xuất bản mã được viết hoàn toàn bởi AI, với một số người thừa nhận họ thậm chí chưa xem xét mã trước khi triển khai. Xu hướng này làm dấy lên lo ngại về sự phụ thuộc ngày càng tăng vào AI

> trong việc lập trình và những hậu quả tiềm ẩn của việc tự động hóa các nhiệm vụ phức tạp mà không có sự giám sát của con người. Trong khi đó, sự kiện Enhanced Games đầu tiên, một cuộc thi thể thao cho phép sử dụng thuốc tăng cường hiệu suất, sẽ diễn ra tại Las Vegas. Sự kiện này phản ánh một sự ám ảnh văn hóa rộng lớn hơn về việc tăng cường và tối ưu hóa, nơi các cá nhân được khuyến khích đẩy ranh giới của hiệu suất con người. Trong thế giới của AI, CEO của Google DeepMind, Demis Hassabis, đã tuyên bố rằng nhân loại đang đứng ở "đồi chân của sự kỳ dị." Thông báo của Google về Gemini for Science, một hệ thống AI có thể thực hiện các dự án nghiên cứu tiên tiến mà không cần sự can thiệp của con người, được coi là một bước tiến hacia mục tiêu này. Tuy nhiên, một số nhà nghiên cứu đang tập trung vào việc phát triển các model thế giới, nhằm tạo ra các hệ thống AI hiểu được môi trường vật lý. Sự phát triển này có ý nghĩa quan trọng về cách AI hiểu thực tại và có thể thay đổi lĩnh vực nghiên cứu AI.

⚡ TIPS & TRICKS CHO DEV

⚡ Tối ưu hóa Retrieval-Augmented Generation

Vấn đề: RAG thường yêu cầu nhiều tài nguyên để đào tạo và triển khai.
Cách làm: Sử dụng các mô hình tiền đào tạo như BERT, RoBERTa để tối ưu hóa quá trình đào tạo RAG. Ví dụ, với câu prompt "Tổng hợp thông tin về AI".
Đánh giá: Hiệu quả cao trong việc giảm thiểu tài nguyên, nhưng yêu cầu kiến thức chuyên sâu về mô hình ngôn ngữ.

⚡ Tích hợp Embeddings vào Semantic Search

Vấn đề: Tìm kiếm семантис truyền thống thường không hiệu quả với dữ liệu phức tạp.
Cách làm: Sử dụng các thư viện như Hugging Face để tạo embeddings cho văn bản và tích hợp vào hệ thống tìm kiếm. Ví dụ, lệnh CLI `python -m transformers` để tải mô hình.
Đánh giá: Cải thiện đáng kể hiệu quả tìm kiếm, nhưng yêu cầu tài nguyên tính toán lớn.

⚡ Áp dụng Semantic Search trong Ứng dụng Thực tế

Vấn đề: Tìm kiếm thông tin liên quan trong cơ sở dữ liệu lớn.
Cách làm: Sử dụng các công cụ như Elasticsearch để triển khai semantic search, với ví dụ prompt "Tìm kiếm thông tin về công ty công nghệ".
Đánh giá: Hiệu quả cao trong việc tìm kiếm thông tin liên quan, nhưng yêu cầu cấu hình và tối ưu hóa hệ thống.

📚 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). Điều này giúp giảm thiểu chi phí vận hành và cải thiện trải nghiệm người dùng. Việc tối ưu hóa LLM cũng giúp tăng tốc độ xử lý và giảm thiểu tài nguyên cần thiết.
3. Ví dụ, việc sử dụng kỹ thuật fine-tuning và LoRA (Low-Rank Adaptation) có thể giúp giảm thiểu kích thước mô hình và cải thiện hiệu năng. Ví dụ code: `model = LLM.from_pretrained('base_model'); model.fine_tune(...)`
4. 💡 Tip: Sử dụng các công cụ như Hugging Face Transformers để tối ưu hóa LLM và giảm thiểu chi phí vận hành. Ngoài ra, cũng nên xem xét việc sử dụng mô hình LLM cục bộ (local LLM) để giảm thiểu phụ thuộc vào dịch vụ đám mây.