BIS - MT · 29/05/2026

🤖 Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

✨ *"If opportunity doesn't knock, build a door."*

↪ Nếu cơ hội không gõ cửa, hãy xây một cánh cửa.

— Milton Berle

💡 *Người chủ động không ngồi chờ cơ hội mà tự tạo ra môi trường và điều kiện để cơ hội xuất hiện.*

📰 TIN TỨC NỔI BẬT

1

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 mới được thiết kế để đơn giản hóa việc tạo ra các ứng dụng đa trợ lý. ADK là một bộ công cụ phát triển phần mềm cho phép các nhà phát triển xây dựng các giao diện trò chuyện và tích hợp chúng với các dịch vụ khác nhau của Google. Với ADK, các nhà phát triển có thể tạo ra các trợ lý có thể tương tác với người dùng thông qua nhiều kênh khác nhau, bao gồm giọng nói, văn bản và giao diện web. Bộ công cụ cung cấp một loạt các thành phần và API đã được xây dựng sẵn, giúp việc xây dựng và triển khai các ứng dụng đa trợ lý trở nên dễ dàng hơn. ADK được xây dựng trên nền tảng Cloud AI của Google và hỗ trợ tích hợp với các dịch vụ khác của Google, chẳng hạn như Dialogflow, Google Cloud Natural Language và Google Cloud Vision. Điều này cho phép các nhà phát triển tận dụng sức mạnh của AI và khả năng học máy của Google để xây dựng các trải nghiệm trò chuyện tinh vi và cá nhân hóa hơn. Bằng cách cung cấp một trải nghiệm phát triển được tối ưu hóa, ADK nhằm mục đích giúp các nhà phát triển xây dựng và triển khai các ứng dụng đa trợ lý dễ dàng hơn, cho phép các trường hợp sử dụng và đổi mới mới trong các lĩnh vực như dịch vụ khách hàng, chăm sóc sức khỏe và giáo dục.

2 LangChain ra mắt công cụ xây dựng agent mã nguồn mở "tiếp cận dễ dàng"

🇬🇧 *LangChain launches "accessible" open source agent builder*

📰 thestack.technology 🔗 Đọc bài viết →

> LangChain đã giới thiệu một công cụ xây dựng tác nhân mã nguồn mở, nhằm giúp các developer dễ dàng tạo và triển khai các tác nhân được hỗ trợ bởi AI. Công cụ này được thiết kế để dễ tiếp cận, cho phép người dùng xây dựng và tùy chỉnh các tác nhân của riêng họ mà không cần kiến thức chuyên sâu về phát triển AI. Công cụ xây dựng mã nguồn mở này được xây dựng trên nền tảng công nghệ hiện có của LangChain, cho phép tạo ra các model AI và ứng dụng phức tạp. Bằng cách cung cấp một giao diện thân thiện với người dùng, công cụ xây dựng tác nhân này dự kiến sẽ đẩy nhanh quá trình phát triển và giảm rào cản gia nhập cho những người muốn tích hợp AI vào dự án của mình. Việc ra mắt công cụ xây dựng tác nhân mã nguồn mở này là một phần trong nỗ lực của LangChain nhằm thúc đẩy việc áp dụng AI và làm cho nó trở nên dễ tiếp cận hơn với đối tượng rộng lớn hơn. Mục tiêu của công ty là trao quyền cho các developer xây dựng các ứng dụng và giải pháp sáng tạo sử dụng AI, thúc đẩy sự đổi mới hơn nữa trong lĩnh vực này.

3 Tăng tốc phát triển với máy chủ Amazon Bedrock AgentCore MCP | 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 MCP là một thành phần chính của nền tảng Amazon Bedrock, cung cấp dịch vụ quản lý để xây dựng, triển khai và quản lý các model AI. Máy chủ Amazon Bedrock AgentCore MCP là một máy chủ hiệu suất cao cho phép các nhà phát triển chạy và đào tạo các model AI với quy mô lớn. Nó cung cấp một môi trường an toàn và có thể mở rộng cho phát triển AI, cho phép các nhà phát triển tập trung vào việc xây dựng và tinh chỉnh các model của họ mà không cần phải lo lắng về cơ sở hạ tầng cơ bản. Với máy chủ Amazon Bedrock AgentCore MCP, các nhà phát triển có thể tận dụng nguồn tài nguyên tính toán khổng lồ và chuyên môn về AI của AWS để tăng tốc quá trình phát triển của họ. Máy chủ này hỗ trợ nhiều framework và

công cụ AI, giúp dễ dàng tích hợp với các workflow và pipeline hiện có. Bằng cách tận dụng máy chủ Amazon Bedrock AgentCore MCP, các nhà phát triển có thể tăng tốc phát triển và triển khai AI của họ, và đưa các model AI của họ ra thị trường nhanh hơn.

4

Các nhà nghiên cứu tự động hóa thiết kế chiến lược suy luận LLM và cắt giảm sử dụng token 69,5%

🇬🇧 *Researchers automated LLM reasoning strategy design and cut token usage by 69.5%*

📰 VentureBeat 🔗 Đọc bài viết →

Các nhà nghiên cứu từ Meta, Google và một số trường đại học đã phát triển AutoTTS, một framework tự động hóa việc thiết kế chiến lược test-time scaling (TTS) cho các mô hình ngôn ngữ lớn (LLMs). TTS nâng cao khả năng của LLMs bằng cách cung cấp thêm tài nguyên tính toán trong quá trình suy luận, cho phép chúng tạo ra nhiều đường lối suy luận hoặc đánh giá các bước trung gian trước khi đưa ra câu trả lời cuối cùng. Tuy nhiên, việc thiết kế chiến lược TTS bằng tay đã từng là một điểm nghẽn, phụ thuộc vào trực giác của con người và dẫn đến sự đánh đổi không tối ưu giữa độ chính xác của mô hình và chi phí tính toán. AutoTTS định nghĩa lại việc thiết kế chiến lược TTS thành một vấn đề tìm kiếm thuật toán, cho phép một mô hình LLM khám phá đề xuất và tinh chỉnh các bộ điều khiển chỉ định cách mô hình phân bổ ngân sách tính toán của mình trong quá trình suy luận. Framework này dựa trên một môi trường phát lại ngoại tuyến để làm cho quá trình tìm kiếm trở nên hợp lý về mặt tính toán. Trong các thử nghiệm, AutoTTS đã giảm thành công việc tiêu thụ token lên đến 69,5% mà không ảnh hưởng đến độ chính xác, vượt qua các thuật toán TTS được thiết kế bằng tay. Framework cũng đã chứng minh khả năng tổng quát hóa của mình đối với các mô hình và nhiệm vụ khác nhau, bao gồm cả một chuẩn mực lý luận tổng quát cấp sau đại học.

5

Sau Orthogonality: Cơ quan đạo đức và sự phù hợp của AI

🇬🇧 *After Orthogonality: Virtue-Ethical Agency and AI Alignment*

📰 The Gradient 🔗 Đọc bài viết →

Bài luận "Sau Tính Chính Orthogonal: Cơ quan Đạo đức - Đức hạnh và Sự Liên kết AI" lập luận rằng hành động hợp lý của con người không được chỉ đạo tới các mục tiêu cụ thể, mà thay vào đó là một phần của mạng lưới các thực hành cấu trúc và thúc đẩy chúng. Tác giả đề xuất

rằng các tác nhân trí tuệ nhân tạo (AI) nên được thiết kế để chia sẻ logic dựa trên thực hành này, thay vì bị thúc đẩy bởi mục tiêu hoặc quy tắc. Cách tiếp cận này là thiết yếu để liên kết AI với các giá trị của con người như minh bạch, hữu ích và vô hại. Tác giả giới thiệu khái niệm "tính hợp lý eudaimonic", được lấy cảm hứng từ ý tưởng về sự thịnh vượng của con người (eudaimonia). Tính hợp lý eudaimonic là một hình thức hoạt động hợp lý đánh giá cao cấu trúc của sự suy xét hơn là kết quả cụ thể. Tác giả lập luận rằng cách tiếp cận này ổn định và an toàn hơn tính hợp lý truyền thống theo hệ quả, ưu tiên mục tiêu hơn phương tiện. Bài luận đề xuất rằng tính hợp lý eudaimonic là một hình thức cơ quan tự nhiên và hiệu quả, và nó được liên kết với các giá trị của con người hơn là các xem xét an toàn AI truyền thống. Tác giả đề xuất rằng các tác nhân AI nên được thiết kế để thúc đẩy sự thịnh vượng của con người theo cách phù hợp với tính hợp lý eudaimonic, thay vì bị thúc đẩy bởi mục tiêu hoặc quy tắc cụ thể.

6

Internet đang được xây dựng lại cho máy móc

🇬🇧 *The internet is being rebuilt for machines*

📰 TechCrunch AI 🔗 Đọc bài viết →

Internet đang trải qua một sự chuyển đổi đáng kể khi các công ty công nghệ thích nghi với sự hiện diện ngày càng tăng của các tác nhân trí tuệ nhân tạo (AI). Những tác nhân này, có thể nhanh chóng truy vấn cơ sở dữ liệu và gọi API, đang tạo ra những thách thức mới cho cơ sở hạ tầng đám mây được thiết kế cho người dùng con người. Amazon Web Services (AWS) đã ra mắt một phiên bản mới của OpenSearch Serverless, một hệ thống tìm kiếm và cơ sở dữ liệu vector được quản lý hoàn toàn, được thiết kế đặc biệt cho các khối lượng công việc của tác nhân. Hệ thống này có thể mở rộng quy mô lên và xuống ngay lập tức để đáp ứng nhu cầu của các tác nhân AI, giảm chi phí cho khách hàng. Sự chuyển dịch sang lưu lượng truy cập được tạo ra bởi máy móc đang tăng tốc, với Cloudflare báo cáo rằng các bot chiếm 31% lưu lượng truy cập HTTP tổng thể trong sáu tháng qua. Các chuyên gia dự đoán rằng lưu lượng truy cập không phải của con người sẽ vượt qua lưu lượng truy cập của con người vào giữa năm 2027. Các nhà cung cấp đám mây khác, bao gồm Microsoft và Databricks, cũng đang tái định vị các dịch vụ của họ để xử lý lưu lượng truy cập của tác nhân AI, nhấn mạnh nhu cầu về cơ sở hạ tầng được thiết kế cho một thế giới ngày càng được các máy móc chiếm lĩnh.

7

MCP trên chế độ mã

🇬🇧 *MCP on Code Mode*

📰 Changelog 🔗 Đọc bài viết →

> Trong một tập gần đây của MCP trên Code Mode, Matt Carey từ Cloudflare thảo luận về khái niệm MCP (Model-Code-Program) và cách nó đã bị hiểu lầm bởi nhiều người. Carey, người làm việc trên Agents SDK và MCP tại Cloudflare, giải thích cách chế độ Code Mode phía máy chủ cho phép một máy chủ MCP duy nhất lộ tất cả 2.500 điểm cuối API của Cloudflare bằng cách sử dụng chỉ 1.000 token ngữ cảnh. Ông cũng đề cập đến bộ nạp Worker động, chạy mã được viết bởi model một cách an toàn trong một V8 isolate, và chia sẻ quy trình làm việc của mình với Claude, một model. Ngoài ra, Carey thảo luận về tầm quan trọng của bộ nhớ trong tương lai của các agent và việc sử dụng một trình bao git, Zaggy, để ngăn chặn đẩy lực lên các kho lưu trữ của mình. Tập này được tài trợ bởi các công ty khác nhau, bao gồm Coder, Tailscale, RWX và Fly.io.

8

Microsoft 365 Copilot nhận được tăng tốc và thiết kế sạch hơn

🇬🇧 *Microsoft 365 Copilot gets a speed boost and cleaner design*

📰 The Verge AI 🔗 Đọc bài viết →

> Microsoft đã ra mắt phiên bản cập nhật của Microsoft 365 Copilot, một trợ lý tập trung vào năng suất được thiết kế để nâng cao trải nghiệm người dùng. Phiên bản Copilot mới có thiết kế sạch sẽ hơn, tải nhanh gấp đôi, cung cấp các phản hồi đáng tin cậy và có cấu trúc hơn, dễ dàng xem xét hơn. Thiết kế lại bao gồm một tính năng gọi là "tiết lộ tiến bộ", trình bày cho người dùng các công cụ và điều khiển phù hợp dựa trên lời nhắc của họ, thay vì hiển thị nhiều tùy chọn cùng một lúc. Ngoài ra, người dùng hiện có thể định dạng văn bản trực tiếp trong hộp lời nhắc của Copilot, hộp này sẽ mở rộng để chứa các đầu vào dài. Phiên bản Copilot cập nhật đang được triển khai trên máy tính để bàn và thiết bị di động, cho phép người dùng truy cập nó trong các ứng dụng Microsoft 365, chẳng hạn như một bảng điều khiển hoặc cửa sổ trò chuyện trong tài liệu, bảng tính hoặc trình chiếu. Cập nhật này nhằm cải thiện tương tác và năng suất của người dùng.

⚡ TIPS & TRICKS CHO DEV

⚡ Quản lý context window

Vấn đề: Giới hạn kích thước context window ảnh hưởng đến hiệu suất AI.
Cách làm: Sử dụng kỹ thuật slicing, chia input thành các phần nhỏ hơn, như ví dụ prompt "tóm tắt văn bản dài 500 từ thành 5 phần".
Đánh giá: Hiệu quả khi xử lý văn bản dài, nhưng cần cân nhắc về độ chính xác.

⚡ Tối ưu hóa long-context

Vấn đề: Xử lý văn bản dài gây ra lãng phí bộ nhớ.
Cách làm: Áp dụng kỹ thuật chunking, chia văn bản thành các phần nhỏ, sau đó xử lý từng phần, như lệnh CLI "python chunking.py --chunk_size 1024".
Đánh giá: Hiệu quả khi xử lý văn bản lớn, giúp giảm thiểu bộ nhớ.

⚡ Tăng cường memory

Vấn đề: Giới hạn bộ nhớ ảnh hưởng đến hiệu suất AI.
Cách làm: Sử dụng kỹ thuật caching, lưu trữ dữ liệu thường xuyên truy cập, như ví dụ prompt "tìm kiếm thông tin về chủ đề X trong bộ nhớ đệm".
Đánh giá: Hiệu quả khi truy cập dữ liệu thường xuyên, giúp giảm thiểu thời gian xử lý.

📚 BÀI HỌC AI HÔM NAY CHO DEV

1. **Tối ưu chi phí & hiệu năng LLM**
Dev cần biết cách tối ưu hóa chi phí và hiệu năng của Large Language Model (LLM) để đảm bảo ứng dụng AI của họ hoạt động hiệu quả và tiết kiệm. Việc này giúp giảm thiểu chi phí vận hành và cải thiện trải nghiệm người dùng.

2. Việc tối ưu hóa LLM liên quan đến việc tinh chỉnh mô hình, chọn cấu hình phù hợp và sử dụng các kỹ thuật như LoRA (Low-Rank Adaptation) để giảm kích thước mô hình mà không ảnh hưởng đến hiệu suất.

3. Ví dụ, khi sử dụng mô hình LLaMA, dev có thể áp dụng kỹ thuật LoRA để giảm kích thước mô hình từ 7B xuống còn 1B, giúp giảm chi phí tính toán và tăng tốc độ xử lý.

4. 💡 Tip hoặc bước tiếp theo: Dev nên bắt đầu bằng việc đánh giá mô hình LLM hiện tại và xác định các phần nào có thể được tối ưu hóa, sau đó áp dụng các kỹ thuật như LoRA hoặc fine-tuning để cải thiện hiệu suất và giảm chi phí.