BIS - MT · 20/05/2026

# 🤖 Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

✨ *"Surround yourself with only people who are going to lift you higher."*

↪ Chỉ bao quanh bản thân bằng những người sẽ nâng bạn lên cao hơn.

— Oprah Winfrey

💡 *Môi trường và người xung quanh ảnh hưởng lớn đến sự phát triển của bạn — chọn lựa cẩn thận ai sẽ là nguồn cảm hứng và động lực cho mình.*

## 📰 TIN TỨC NỔI BẬT

1

### OpenAI ra mắt Chrome Extension Codex hỗ trợ tác vụ web development nền

🇬🇧 *OpenAI Launches Codex Chrome Extension for Background Web Development Tasks*

📰 Technobezz 🔗 Đọc bài viết →

> OpenAI đã ra mắt một Chrome extension mới có tên Codex, được thiết kế để hỗ trợ các tác vụ web development chạy nền. Extension này tận dụng công nghệ Codex của OpenAI, sử dụng AI để tạo code và tự động hóa các tác vụ lặp đi lặp lại. Với Codex, các developer có thể thực hiện các tác vụ như code completion, debugging và optimization mà không cần rời khỏi workflow hiện tại. Extension này nhằm mục đích đơn giản hóa web development bằng cách tự động hóa các tác vụ thường ngày, giúp các developer tập trung vào các tác vụ cấp cao hơn và cải thiện năng suất. Codex có thể được sử dụng cho nhiều tác vụ khác nhau, bao gồm code refactoring, sửa bug và triển khai feature. Extension này hiện có sẵn để tải xuống trên Chrome Web Store và có thể được tích hợp vào các workflow development hiện có. Bằng cách tự động hóa các tác vụ nền, Codex đặt mục tiêu giảm thời gian và công sức cần thiết cho web development, giúp các developer dễ dàng xây dựng và duy trì các web application phức tạp hơn. Việc ra mắt Codex là bước phát triển mới nhất trong nỗ lực của OpenAI nhằm tận dụng công nghệ AI để cải thiện hiệu quả và năng suất của software development.

2

### AWS open-source MCP server cho Bedrock AgentCore để tinh gọn phát triển AI agent

*AWS Open-Sources an MCP Server for Bedrock AgentCore to Streamline AI Agent Development*

MarkTechPost 🔗 Đọc bài viết →

> Amazon Web Services (AWS) đã open-source một MCP (Multi-Cloud Platform) server cho Bedrock AgentCore. Động thái này nhằm mục đích đơn giản hóa quá trình phát triển AI agent. MCP server được thiết kế để hoạt động liền mạch với Bedrock AgentCore, một framework để xây dựng và triển khai AI agent. Việc open-source MCP server được kỳ vọng sẽ tinh gọn quá trình phát triển AI agent bằng cách cung cấp một platform tiêu chuẩn để testing và deployment trên nhiều môi trường cloud khác nhau. Điều này có thể giúp giảm độ phức tạp và thời gian cần thiết để các developer triển khai AI agent trên các cloud platform khác nhau. MCP server hiện đã có sẵn để sử dụng công khai, cho phép các developer truy cập và đóng góp vào codebase. Cách tiếp cận open-source này có thể thúc đẩy sự hợp tác và đổi mới trong cộng đồng AI development, cuối cùng dẫn đến việc triển khai AI agent hiệu quả và năng suất hơn. Bằng cách cung cấp một platform tiêu chuẩn, AWS đang giúp đẩy nhanh quá trình phát triển AI agent và các application của chúng.

## 3 Xây dựng AI 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 tutorial về cách xây dựng một AI agent phân tích tài chính thông minh sử dụng LangGraph và Strands Agents. AI agent này được thiết kế để phân tích financial data và cung cấp insight cho người dùng. Tutorial này sử dụng LangGraph, một NLP library, để xử lý và hiểu financial data. Strands Agents, một low-code platform, được sử dụng để xây dựng và deploy AI agent. AI agent có thể được train để phân tích nhiều nguồn financial data khác nhau, bao gồm các bài báo tin tức, báo cáo tài chính và xu hướng thị trường. Nó cũng có thể được tích hợp với các AWS service khác, như Amazon Comprehend và Amazon SageMaker, để tăng cường khả năng của mình. AI agent có thể cung cấp cho người dùng phân tích tài chính và insight theo thời gian thực, giúp họ đưa ra các quyết định sáng suốt. Tutorial này cung cấp hướng dẫn từng bước về cách xây dựng và deploy AI agent phân tích tài chính thông minh sử dụng LangGraph và Strands Agents. Nó bao gồm các chủ đề như thiết

lập environment, xây dựng AI agent và tích hợp nó với các AWS service khác. Bằng cách làm theo tutorial này, người dùng có thể tạo ra một AI agent phân tích tài chính mạnh mẽ có thể giúp họ đi trước thị trường.

4

## AGI không phải là Multimodal

🇬🇧 *AGI Is Not Multimodal*

📰 The Gradient 🔗 Đọc bài viết →

Những tiến bộ gần đây trong các generative AI model đã khiến một số người tin rằng Artificial General Intelligence (AGI) sắp xuất hiện. Tuy nhiên, những model này xuất hiện không phải do các giải pháp được suy nghĩ kỹ lưỡng, mà là do chúng có thể scale hiệu quả trên hardware hiện có. Cách tiếp cận multimodal, liên quan đến việc tối ưu hóa các modular network khổng lồ cho nhiều modality khác nhau, được

5

## Điều gì khiến một công việc trở nên nhàm chán, bẩn thỉu hay nguy hiểm?

🇬🇧 *What Makes a Job Dull, Dirty, or Dangerous?*

📰 IEEE Spectrum 🔗 Đọc bài viết →

Các nhà nghiên cứu từ Viện RAI đã tái định nghĩa khái niệm "nhàm chán, bẩn và nguy hiểm" (DDD) cho lĩnh vực robot, nhằm hiểu rõ hơn về các loại nhiệm vụ hoặc công việc mà robot có thể hữu ích. Định nghĩa truyền thống về công việc DDD, chẳng hạn như lao động thể chất lặp đi lặp lại trong nhà máy, không đơn giản như vẻ ngoài. Một nghiên cứu đã phân tích các ấn phẩm về robot từ năm 1980 đến 2024 và phát hiện rằng chỉ một tỷ lệ nhỏ cung cấp định nghĩa và ví dụ rõ ràng về công việc DDD. Để giải quyết vấn đề này, các nhà nghiên cứu đã xem xét tài liệu khoa học xã hội trong các lĩnh vực khác nhau để phát triển các định nghĩa tinh tế hơn về công việc "nhàm chán", "bẩn" và "nguy hiểm". Họ phát hiện ra rằng các loại công việc này bị ảnh hưởng bởi các yếu tố xã hội, kinh tế và văn hóa. Ví dụ, công việc "bẩn" không chỉ liên quan đến sự bẩn thỉu về thể chất, mà còn về sự kỳ thị xã hội, với các công việc như nhân viên quản giáo và đại lý thu nợ được coi là "bẩn" do liên quan đến các nhóm bị kỳ thị. Các nhà nghiên cứu cũng nhấn mạnh tầm quan trọng của việc xem xét phương pháp thu thập và báo cáo dữ liệu khi đo lường mức độ nguy hiểm của một

nhiệm vụ hoặc công việc, vì các chấn thương nghề nghiệp thường bị báo cáo thấp và không được phân chia theo các đặc điểm như giới tính và tình trạng việc làm. Việc tái định nghĩa công việc DDD này có ý nghĩa đối với lĩnh vực robot, nhấn mạnh cơ hội can thiệp vào các lĩnh vực ít rõ ràng hơn và cải thiện an toàn lao động cho các nhóm bị thiệt thòi.

6

## OlmoEarth v1.1: Dòng model hiệu quả hơn

🇬🇧 *OlmoEarth v1.1: A more efficient family of models*

📰 Hugging Face Blog 🔗 Đọc bài viết →

Các nhà nghiên cứu tại Allen AI đã phát hành OlmoEarth v1.1, một họ model mới được thiết kế để tăng hiệu quả trong việc xử lý hình ảnh vệ tinh. Các model cập nhật này cắt giảm chi phí tính toán lên đến 3 lần trong khi vẫn duy trì hiệu suất trên các chuẩn mực và nhiệm vụ nghiên cứu khác nhau. Sự cải thiện này được đạt được bằng cách giảm độ dài chuỗi của các model, điều này giảm đáng kể chi phí chạy model. Các model OlmoEarth dựa trên kiến trúc transformer, một kiến trúc thống trị trong học máy, và được sử dụng để xử lý dữ liệu cảm biến từ xa. Để xử lý dữ liệu cảm biến từ xa, các model đầu tiên chuyển đổi nó thành một chuỗi token. Hai yếu tố chính kiểm soát hiệu quả trong các model dựa trên transformer: kích thước model và độ dài chuỗi token. Các nhà nghiên cứu đã phát hiện ra rằng việc giảm độ dài chuỗi token có thể dẫn đến tiết kiệm chi phí đáng kể. Họ cũng đã khám phá các biểu diễn token khác nhau, chẳng hạn như chia dữ liệu thành các bản vá dựa trên độ phân giải hoặc gộp độ phân giải vào một token duy nhất. Mặc dù việc gộp độ phân giải dẫn đến tiết kiệm đáng kể, nhưng nó dẫn đến giảm hiệu suất đáng kể. Các nhà nghiên cứu đưa ra giả thuyết rằng việc tách các băng tần Sentinel-2 thành các token riêng biệt giúp OlmoEarth dễ dàng hơn trong việc mô hình hóa các mối quan hệ giữa băng tần quan trọng.

7

## llm-gemini 0.32

🇬🇧 *llm-gemini 0.32*

📰 Simon Willison 🔗 Đọc bài viết →

Một phiên bản mới của plugin llm-gemini, phiên bản 0.32, đã được phát hành. Cập nhật này có khả năng mang lại các cải tiến và sửa lỗi cho plugin, được sử dụng để làm việc với các mô hình ngôn ngữ lớn

(LLMs). Việc phát hành llm-gemini 0.32 diễn ra sau khi phát hành Gemini 3.5 Flash, một cập nhật đáng kể cho plugin Gemini. Phiên bản mới này có thể cung cấp chức năng nâng cao và khả năng tương thích với các LLM khác nhau. Đối với những người quan tâm đến việc cập nhật các phát triển mới nhất về LLMs, tác giả của plugin đang cung cấp một bản tóm tắt email được tài trợ. Bằng cách đóng góp 10 đô la mỗi tháng, người đăng ký sẽ nhận được một lựa chọn được biên tập của các tin tức và tiến bộ LLM quan trọng nhất trong tháng qua. Các chi tiết cụ thể về các thay đổi và cải tiến trong llm-gemini 0.32 không được cung cấp trong thông tin có sẵn.

8

## Tại sao sandboxing OpenClaw không ngăn chặn được data exfiltration

🇬🇧 *Why sandboxing OpenClaw doesn't stop data exfiltration*

📰 BD Tech Talks 🔗 Đọc bài viết →

Một nghiên cứu bảo mật gần đây đã tiết lộ rằng việc cô lập (sandboxing) alone không thể ngăn chặn việc lấy cắp dữ liệu từ các tác nhân AI tự động như OpenClaw. Mặc dù việc chứa (containerization) và các hộp cát ảo (virtual sandboxes) cách ly việc thực thi độc hại khỏi máy chủ, các nhà nghiên cứu đã tìm thấy các điểm yếu trong môi trường cô lập của Nvidia, NemoClaw, cho phép các tác nhân độc hại thao túng tác nhân để rò rỉ dữ liệu hoặc viết lại các lệnh của nó. Nghiên cứu, được thực hiện bởi công ty bảo mật Lasso, đã chứng minh hai vector tấn công riêng biệt: đầu độc phụ thuộc (dependency poisoning) và đầu độc cấu hình tác nhân (agent configuration poisoning). Trong cuộc tấn công đầu tiên, các tác nhân độc hại đã xuất bản các gói có payload bị che giấu đọc các tệp cấu hình nhạy cảm, trong khi trong cuộc tấn công thứ hai, họ đã đầu độc các tệp cấu hình của tác nhân để lấy cắp dữ liệu. Các nhà nghiên cứu đã sử dụng các kỹ thuật như mã hóa emoji để vượt qua các báo động và bộ lọc bảo mật tĩnh. Những phát hiện này nhấn mạnh hạn chế của các biện pháp bảo mật truyền thống trong việc bảo vệ các hệ thống lưu trữ các tác nhân AI tự động, vì đường dẫn thực thi của chúng được xác định động bởi văn bản mà chúng đọc.

⚡ TIPS & TRICKS CHO DEV

### ⚡ Tối ưu hóa mã

**Vấn đề:** Mã nguồn không tối ưu, gây chậm và tiêu tốn tài nguyên.

**Cách làm:** Sử dụng GitHub Copilot để giúp tối ưu hóa mã, ví dụ: "Optimize this code for performance".
**Đánh giá:** Hiệu quả cao, nên dùng khi cần cải thiện hiệu suất mã nguồn.

### ⚡ Tự động hoàn thành mã

**Vấn đề:** Gây mất thời gian khi phải tự viết mã từ đầu.
**Cách làm:** Sử dụng GitHub Copilot với prompt "Complete this function" để tự động hoàn thành mã.
**Đánh giá:** Tiết kiệm thời gian, nên dùng khi cần hoàn thành mã nhanh chóng.

### ⚡ Xác định lỗi mã

**Vấn đề:** Khó xác định lỗi trong mã nguồn phức tạp.
**Cách làm:** Sử dụng Aider với lệnh "Debug this code" để xác định lỗi mã.
**Đánh giá:** Hiệu quả cao, nên dùng khi cần xác định lỗi trong mã nguồn nhanh chóng.

---

📚 BÀI HỌC AI HÔM NAY CHO DEV

1. **Tối ưu chi phí & hiệu năng LLM**
2. Việc tối ưu hóa chi phí và hiệu năng của mô hình ngôn ngữ lớn (LLM) là rất quan trọng vì nó giúp giảm thiểu chi phí tính toán và tăng tốc độ xử lý. Điều này đặc biệt quan trọng khi triển khai LLM trong các ứng dụng thực tế. Dev cần biết cách tối ưu hóa LLM để đảm bảo hiệu suất và tiết kiệm chi phí.
3. Ví dụ, sử dụng kỹ thuật fine-tuning và LoRA (Low-Rank Adaptation) có thể giúp giảm thiểu kích thước mô hình và tăng tốc độ xử lý. Ví dụ code: `model = transformers.AutoModelForSequenceClassification.from_pretrained("bert-base-uncased", num_labels=8)`
4. 💡 Tip: Sử dụng thư viện như Hugging Face Transformers để tối ưu hóa LLM và triển khai trong các ứng dụng thực tế.