Mar 15, 2025

LLM Agents Study Session

Summary

Yatharth Piplani presented the Model Context Protocol (MCP), a standardized system for connecting LLMs with tools, highlighting its advantages, limitations, implementation, and future directions, including potential smartphone applications and a "meta MCP server." Following the presentation, participants discussed further sessions on agent tracing and observability within the MCP ecosystem.

Details

- **Meeting Time Confusion:** Initial confusion arose regarding the meeting time due to a one-hour time difference between the US and India. Arunesh Mishra and Shristi Gautam resolved the issue by extending the meeting time by half an hour. They attributed the confusion to multiple parties.
- **Daylight Saving Time:** The discrepancy in meeting times was caused by the recent shift to daylight saving time in the US, resulting in a one-hour time difference. Arunesh Mishra, Shristi Gautam, and Yatharth Piplani discussed adjusting future meeting times to accommodate the change.
- **Research Project Collaboration:** Arunesh Mishra announced their return to the US and expressed interest in collaborating on a research project, potentially involving the creation of a technologically advanced product. Yatharth Piplani also expressed interest in collaborating and suggested a separate, unrecorded meeting to discuss ideas.
- **Model Context Protocol (MCP) Overview:** Yatharth Piplani presented on the Model Context Protocol (MCP), describing it as an open protocol standardizing how applications provide context to Large Language Models (LLMs). They highlighted MCP's ability to allow LLMs to use various tools, such as search

engines and custom APIs. They also noted that it addresses the lack of a standardized way to connect LLMs with tools.

- **MCP Advantages and Limitations:** Yatharth Piplani explained MCP's advantages, including standardized interfaces, modularity, and extensibility. They also discussed limitations, such as the unpredictability of user behavior and the potential for misuse of tools like APIs with sensitive data. They emphasized the client-server model of MCP, where the client requests tool usage and receives responses from the server.
- **MCP Implementation and Examples:** The presentation included practical examples of MCP implementation, such as building a custom Google Trends server and using MCP to create GitHub repositories. Yatharth Piplani demonstrated how clients like Cursor integrate with MCP servers. They showed a live demonstration of an MCP client connection and notification system.
- **Comparison to Traditional API Use:** Yatharth Piplani compared MCP to traditional API usage, highlighting MCP’s advantages in terms of plug-and-play functionality, modularity, and easier discovery. They also discussed the limitations of traditional APIs, particularly regarding versioning inconsistencies, rate limits, and error handling. They mentioned Hyrum's Law as a relevant concept in understanding the challenges of API usage and MCP's attempts to mitigate them.
- **MCP Client-Server Interaction:** Yatharth Piplani detailed the client-server communication in MCP, explaining how clients send requests and receive responses from the server. They described the process of initializing a server, handling requests, and ensuring connection readiness. The presentation included details on client-side implementation, including establishing connections, processing queries, and handling responses.
- **MCP Overview and GitHub Integration:** Yatharth Piplani explained the Model Context Protocol (MCP), a system for building and integrating tools. They demonstrated using GitHub as a tool, showcasing the creation of repositories and issues via a simple server and commands. They emphasized the importance of fine-grained access tokens to limit the LLM's access to GitHub. The use of commands for various operating systems was also noted.
- **MCP Tool Use and Automation:** Piplani demonstrated the automation capabilities of MCP, using the `cursor` tool to create a repository, add a README, and create an issue. They highlighted the "YOLO" mode for increased automation

but cautioned against its risks. They further discussed the potential for broader automation using MCP beyond the presented examples.

- **MCP Ecosystem and Future Directions:** Piplani and Shristi Gautam reflected on the growth of the MCP ecosystem, noting its evolution from its early stages. Arunesh Mishra suggested the development of a "meta MCP server" to manage other MCP servers, and they discussed potential applications on smartphones. The potential for MCP to replace traditional API programming was also raised.
- **MCP Agents and Libraries:** Discussions included the development of MCP agents and associated libraries, such as the `mcp-agents` Python library which simplifies agent creation and MCP interaction. The potential for agents to call other agents through MCP was discussed.
- **MCP Inspector and Observability:** Shristi Gautam demonstrated the MCP inspector, a tool for troubleshooting and testing custom MCPs. They and Piplani discussed the need for further sessions on agent tracing and observability within the MCP ecosystem. The inspector was described as a tool that opens in a localhost and allows for testing MCPs.

Suggested next steps

- ☐ Arunesh Mishra will crop the meeting recording by one hour.
- ☐ Arunesh Mishra and Yatharth Piplani will schedule a separate meeting to discuss research project ideas.