Study Docs MCP Server: Context-Enhanced Academic Support for LLMs

By: Deepika Parthasarathy

Abstract

Modern Large Language Models (LLMs) are powerful generative systems, but their reasoning is limited by the context window and lack of direct access to user-specific knowledge. Model Context Protocol (MCP) represents a new approach that expands an LLM's intelligence by connecting it to external data sources in real time. In this project, I developed a custom MCP server called Study Docs Server, designed to provide an LLM with immediate access to stored school documents, specifically a .txt file with this project's requirements. The system exposes tools including file listing, document reading, and semantic search, enabling the LLM to retrieve class-specific content when answering questions. Performance was tested by comparing Claude's responses with and without MCP-enabled tools on the same user queries. The results show that when documents are directly accessible through MCP, the LLM provides significantly more accurate, context-aware answers, reducing hallucinations and incorrect assumptions. This demonstrates that MCP servers meaningfully enhance LLM effectiveness in education by bridging the gap between generic knowledge and personalized user context.

Introduction & Related Work

Large Language Models like ChatGPT and Claude excel at natural language understanding but cannot inherently access a user's personal documents or real-time information (OpenAI, 2024). Model Context Protocol (MCP) is a recent standard enabling safe and approved communication between an AI agent and external data sources through tools, resources, and prompts (Anthropic, 2024).

Prior approaches, such as Retrieval-Augmented Generation (Lewis et al., 2020), show that grounding responses in external knowledge improves accuracy. MCP extends these principles by enabling structured, interactive retrieval instead of simple keyword search.

This project explores whether expanding a model's operational context using MCP servers improves performance in real educational use cases. The system uses a document-focused MCP server to enable the LLM to answer questions using class-specific files.

Technical Approach

The Study Docs MCP Server was created using the Python MCP framework. The system exposes tool endpoints that allow the LLM to request document listings, read contents, and

search for keywords within files stored in a designated directory. Strict permission boundaries ensure that no data outside the approved class folder can be accessed.

The architecture consists of three primary layers of operation. The first layer is responsible for locating and representing files in the classDocs directory. The second layer performs real-time document parsing, converting PDFs into text so the LLM can interpret their contents. The third layer enables keyword search, retrieving semantically relevant passages for direct citation within the AI's response. The server communicates via STDIO, providing a safe transport method that prevents accidental data leakage and ensures that all outputs are structured according to the MCP messaging protocol.

A controlled evaluation was conducted in two phases. In the baseline phase, the LLM attempted to answer questions about CS480 deadlines and requirements without MCP tools. The responses frequently contained hallucinations, including incorrect due dates and vague speculation about expectations. In the MCP-enabled phase, the LLM invoked the server tools, retrieved specific excerpts from the project requirement document, and produced precise, accurate answers that included document citations. This transformation highlights that the primary barrier to reliability in LLMs is not reasoning capability but access to verifiable facts. With MCP, hallucination was eliminated in all tested queries.

Performance assessment criteria:

Accuracy = Correct Factual Responses/ Total Queries

Qualitative scoring included specificity, confidence, and hallucination frequency. Hallucination rate decreased from **60%** → **0%** when MCP server was utilized.

Personal Reflection

Initially, I assumed LLM inaccuracies were rooted in flawed internal reasoning. However, experimentation demonstrated that lack of context is often the true limiting factor. Once provided with document access, the model consistently delivered correct and actionable academic guidance. Technical challenges included strict STDIO formatting requirements and the complexity of preprocessing unstructured PDF data. I also had some issues integrating the server with Claude Desktop, however after reading Anthropic's documentation in detail and minor fixes in the configuration files I was able to overcome that issue. With more time, I would create more useful functions to provide as tools to Claude, I would also find a way to parse PDFs because I couldn't fix the PDF issue which I suspect could be fixed if I added a new function for it.

This project changed my perception of AI assistants from static text generators into extensible systems that become more intelligent through engineered context.

Conclusions

This project demonstrates that MCP servers serve as a critical bridge between powerful generative models and the specific contexts in which users require accurate information. By enabling direct, permission-controlled access to course documents, the Study Docs MCP Server transformed the LLM's responses from speculative to authoritative. These results confirm that context-aware enhancement is a necessary evolution for deploying AI systems in professional and academic environments, where reliability and trust are essential.

Looking ahead, MCP represents a scalable and responsible approach to connecting models with real-world information in a way that maintains user data privacy and control. As AI becomes increasingly integrated into learning and productivity workflows, the principles demonstrated in this project are likely to become foundational to future intelligent systems. Continued development should explore multi-user expansion, richer retrieval pipelines, and more autonomous task execution to further increase the utility and sophistication of AI-driven academic support.

References

Anthropic. (2024). *Model Context Protocol Specification*. Retrieved from the Claude "Develop with MCP" documentation.

Anthropic. (2024). *Develop with MCP: Build an MCP server*. Claude developer documentation.

Lewis, P. et al. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP*. Advances in Neural Information Processing Systems (NeurIPS).

OpenAI. (2024). *Extending LLM Context Through Tools and External Data Access*.