Sample Technical Report: AI Agent Development

Lalith Ashish

June 2025

Abstract

This report explores the development of a multi-tool AI agent designed to handle complex tasks using natural language processing. The agent integrates tools like web search and code execution, offering insights into task planning, tool selection, and memory management for enterprise applications.

1 Introduction

Artificial Intelligence (AI) agents are transforming enterprise workflows by automating complex tasks. This report details the design and implementation of a multi-tool Large Language Model (LLM) agent, capable of interpreting user queries, selecting appropriate tools, and delivering coherent responses. The project aligns with industry trends toward autonomous systems, as seen in tools like LangChain and OpenAIs GPT models. The agent is built to break tasks into subtasks, execute them using tools like a calculator or summarizer, and store results for future use. This report covers the methodology, challenges, and findings from the development process.

2 Methodology

The agent was developed using Python and LangChain, with OpenAIs GPT-4o-mini as the LLM backend. Four tools were integrated: a web search via Serper API, a math calculator using Pythons eval, a Python code executor, and a document summarizer. The agent employs a task planner to decompose queries into subtasks, a tool router to select tools based on keywords, and a memory module using JSON for long-term storage. Testing involved three complex queries, such as calculating trip costs and summarizing AI news, to evaluate performance.

3 Findings

The agent successfully handled all test queries, achieving 100% task success. It correctly used the calculator for math tasks (e.g., computing $700 for a trip) and web search for factual queries (e.g., Tokyos population). The summarizer condensed texts effectively, producing 50-word summaries. Challenges included occasional API errors, addressed via a retry mechanism. The agents modular design ensures scalability, making it suitable for enterprise applications requiring automation and data retrieval.

4 Conclusion

The multi-tool LLM agent demonstrates robust capabilities in task automation and tool integration. Future improvements could include advanced error handling and support for larger documents. This project highlights the potential of AI agents in enhancing productivity across industries.