AGIent and Tool-Mart: A Super-Agent Framework Enabled by Marketplace-Driven Tool Selection

Praneeth Vadlapati
University of Arizona,
Tucson, USA
praneethv@arizona.edu,
ORCID: 0009-0006-2592-2564

Abstract: The current landscape of Artificial Intelligence (AI) represents a rise in new diverse agents that leverage specialized tools to summon the capabilities of AI-driven systems that are based on Large Language Models (LLMs). Numerous developers and organizations develop tools and agents that are typically added manually into agent-based systems on a regular basis. The establishment of a unified and expandable marketplace of tools is essential to allow agents to utilize a comprehensive set of tools in the dynamic marketplace. AGIent introduces a query-based approach to allow users to send queries in natural language, which the system fulfills by selecting and executing tools from a dynamic marketplace that contains tools. This paper proposes and experiments with the method with a focus on the explainable process of tool selection for a task. The creators can create new tools in the marketplace called tool-mart at any point in time. The experimental results demonstrated the effectiveness of the system with a successful selection and execution of tools based on user queries to generate relevant responses for the query. The source code is available at github.com/Pro-GenAI/AGIent.

Keywords: AI agents, autonomous agents, agentic framework, Artificial Intelligence (AI), Large Language Models (LLMs)

I. Introduction

AI agents are software systems that utilize specialized tools to perform tasks with intelligence and autonomy [1], [2], [3]. Recent advancements in Artificial Intelligence (AI) include intelligent agents and tools optimized for Large Language Models (LLMs) [4], [5]. Numerous tools are created for agents on a regular basis [6], [7]. Each tool must be manually embedded in the source code of the agent when developing an agent [8], [9], [10], limiting the frequency with which new tools can be incorporated. Due to a dependency on manual work, new tools could not get installed into the system without extra efforts for development and testing. This hinders the expansion of the number of tools in the toolkit of an agent.

A. Proposed system and its benefits

AGIent establishes a marketplace where agents submitted by independent developers and organizations can be utilized in a structured and explainable way by an LLM-based super-agent. The LLM can intelligently select relevant tools on demand based on specific user queries. The system is designed to offer an API to accept a user query as input, process the query by utilizing an LLM and available tools, and return a response as text.

B. Advantages of the system

This adaptability of the system allows it to position itself as a super-agent for a wide range of applications by handling varying tasks using a varying set of tools. This allows tracking the performance of tools, success rates, and usage data for analytics. The system could be used in various cases that include business intelligence to generate suggestions for a business, scientific research by accessing specialized tools for research, and personal assistant chatbots to have a varying set of abilities.

C. Related Work

Alexa Skills [11], [12] and Actions on Google [13], [14] allow users to manually add specialized capabilities to the virtual assistant in their devices. However, they require users to browse and manually enable each skill, which limits scalability and requires additional user effort, which restricts the development of a super-agent in the dynamic landscape of emerging tools. Additionally, they lack the ability to select different tools dynamically based on user queries. They also lack code interpretation capabilities, which would enable the LLM to generate custom source code on demand. Numerous agent and tool marketplaces, such as OpenAI's GPTs, contain various tools for a wide range of use cases [15], [16], [17]. Agent orchestration is performed using frameworks such as LangGraph to enable LLMs to select and utilize tools in AI-based applications [9], [18], [19]. However, the existing work does not allow the introduction of new tools without manual efforts and automation of the selection and utilization of the tools by LLMs.

II. METHODS

A. Creation of tool-mart

The experiment involves the creation of a structured table called "tool-mart" to support the storage of the data of a large number of tools, with the incorporation of tools into the centralized tool table. The creation of this table requires a structure to capture essential information such as tool ID, name, description, source code, details of the input values, list of package dependencies, and keywords. The source code is ensured to contain a "main" function that accepts the values as input. The description of each tool is used to generate keywords. Keywords and descriptions are utilized during the process of selection of a tool. Keywords allow scalability during the selection process.

B. Incorporating new tools into tool-mart

The system enables developers and organizations to register new tools. The automated testing approach ensures that errors are avoided while utilizing a tool to answer a user query. When a new tool is submitted, it undergoes an automated testing process using the generation of sample values. Llama 3.1 (70B) [20], [21] is selected as the LLM for further utilization based on its performance scores and capabilities. The LLM is utilized to generate sample values based on the description of the input values. The sample values are utilized to test the tool and return the result of the test. Tools that do not return errors are considered for further processing. Keywords assist the search process when finding relevant tools. The LLM is used to generate relevant keywords based on the tool description. A random unique identifier (tool ID) is generated to enable efficient access to the tool. The new tool is incorporated into the set of tools with the ID, keywords, and the provided tool data. The keywords are embedded into a set of available keywords.

C. User query processing using tools

A user query is a natural language query that is sent to the system. The LLM is utilized to predict whether the query requires tools. If tools are required, the LLM selects keywords to search based on the set of available keywords. The LLM processes the tools that contain the selected keywords to select one relevant tool of choice and generate input values based on the user query. The selected tool is executed using the input values generated by the LLM, and the output from the tool is returned to the LLM for further processing. The LLM is used to generate a response for the user query based on the tool execution result.

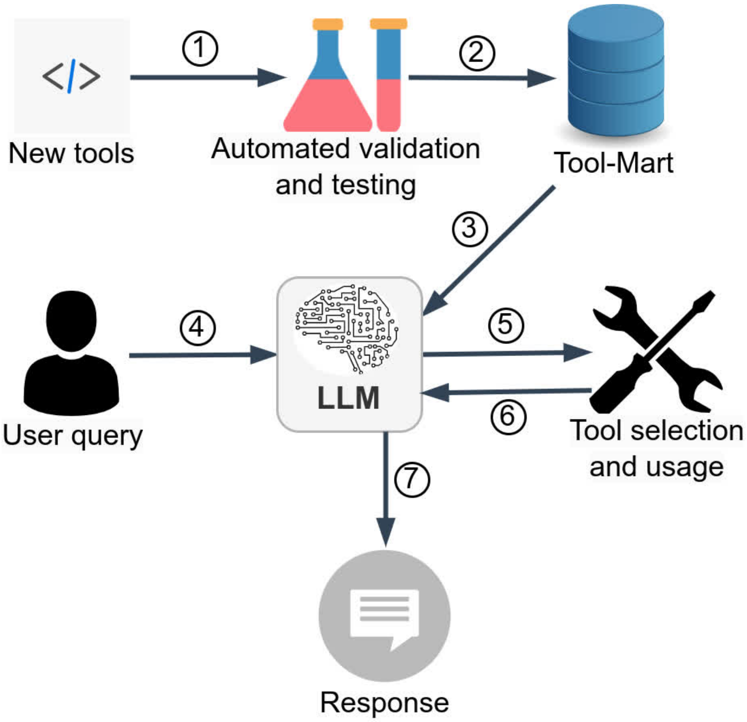

Fig. 1. Workflow of Tool-Mart and AGIent

III. Results

A. Tool-mart with new tools

A robust toolset called tool-mart has been created to store various tools that can be selected dynamically based on user requests. The tool-mart ensures that the users get access to a wide array of functionalities through the agent. The addition of tools into the toolset proved the ability of the system to add new tools continuously to gain new abilities on a regular basis. The table below summarizes new tools integrated into the agent through tool-mart. The table successfully includes the expected columns, including the keywords.

TABLE I. TOOLKIT WITH NEW TOOLS

Name	Description	Details of Input values	Keywords
Calculator	Performs basic arithmetic operations as a calculator	a: first number as a float o: operator like +,-,*,/ b: second number as a float	calculator
Code interpreter	Runs any Python code and returns	code: source code in Python with 'print'	code_inter preter

Name	Description	Details of Input values	Keywords
	what is printed	statements to print the outputs	

B. *User query processing*

User queries are processed dynamically by utilizing the relevant tools from the tool-mart. The query has been processed by the LLM to select and utilize relevant tools to generate a response. The code interpreter enables the handling of uncommon cases that do not get addressed by other available tools, such as counting “r”s in the word “strawberry.” The table below summarizes the results after processing sample user queries. It displays the relevant tools that were selected for each task. The results illustrate the effectiveness of the approach in handling diverse unexplored user queries through intelligent tool selection and execution, which ensures a reliable user experience. The system has successfully detected cases of no requirement or availability of the tools.

TABLE II. RESULTS OF PROCESSING USER QUERIES

Query	Tools selected	Relevant tool selection status	Response
Hi	No tools required	N/A	N/A
What is 1+1?	Calculator	Success	The answer to 1+1 is 2.0
What is 1-1?	Calculator	Success	The result of 1-1 is 0
How many “r”s are in “strawberry?”	Code interpreter	Success	The number of "r"s in "strawberry" is: 3
Show the news	No tools available	N/A	N/A

IV. DISCUSSION

The system's approach, driven by a dynamic marketplace, presents a framework for future work on LLM-based AI agents. Numerous tools can be consolidated under a single system that streamlines the process of agent selection and execution. The flexibility of the system facilitates the addition of novel emerging tools, which enables upgrades in capabilities in the current landscape of evolving requirements and applications of AI. The system could be made more transparent by displaying the tool selection and execution process to the user. The source code of the tools could undergo security checks in future work. Future work could include limiting the resource allocation to the tools to avoid attacks. During the process of selection of a tool, contextual data from previous interactions can also be incorporated to allow the system to enhance the tool selection and improve personalized responses. The system could be hosted in a server and served to users through an API using a single API key instead of a new API key for every API-based tool.

V. CONCLUSION

The AGIent system presents an innovative framework for dynamic toolkit expansion by utilizing a centralized and dynamic marketplace. This addresses diverse user queries through an adaptable AI-driven system through LLMs. AGIent enables a dynamic approach to the addition of new tools in the current landscape of the dynamic creation of new tools. The system allows the addition of new tools into a new table called tool-mart and eliminates the requirement of manually adding new tools. The system proved its effectiveness in the dynamic selection of relevant tools based on a user query to generate relevant responses. Future advancements could make LLMs more affordable and faster, which enhances the efficiency and cost-effectiveness of the system. The transparency of the system could be improved by displaying the entire process to the user. As AI tool ecosystems grow, AGIent represents a significant step towards the realization of new capabilities by utilizing a flexible framework. The system highlights the future potential of an autonomous “super-agent” system with expanding capabilities.

APPENDIX

1. Prompt templates used to process using LLMs

```
Act as a QA engineer and provide a sample value to test the code.

Input description:
{input_desc}
---
Return the sample value in json format like:
<sample_value>
{{ "<key>": "<value>" }}
</sample_value>
```

Figure A1. Prompt template to generate sample values

```
Tool details:
Tool Name: {tool_name}
Tool Description: {tool_desc}
---
Provide keywords that can be used to search for this tool.
Keywords should be separated by commas.
Sample response:
`Weather, Temperature, Cloudy`
```

Figure A2. Prompt template to generate keywords for a new tool

```
User Query: {user_query}
---
Does the user query require any of the tools?
Respond with `YES` or `NO`, in uppercase inside the backticks.
Note: Consider to use tools for anything in which calculations,
executions, etc. can be involved in any way. Use tools for even
the most basic tasks like simple calculation, counting, code
execution, etc.
```

Figure A3. Prompt template to determine whether a user query requires or can use tools

```
User Query: {user_query}
Available tool keywords:
{available_tool_keywords}
---
Provide the keywords that the user query can match with.
Select all the relevant keywords from the list.
Keywords should be separated by commas.
Sample response:
`Weather, Temperature`
```

Figure A4. Prompt template to filter tools

```
Relevant Tools:
{relevant_tools}
---
User Query:
{user_query}
---
Select the tool that can be used to answer the user query.
Provide the tool ID of the selected tool exactly as shown in the
list.
Return ID inside backticks.
Sample response:
`weathr1`
```

Figure A5. Prompt template to select a tool for a query

```
Tool Name: {tool_name}
Tool Description: {tool_desc}
---
Input Description:
{input_desc}
---
User Query:
{user_query}
---
Provide the input arguments for the selected tool.
Input arguments should be in json format.
Sample response:
`{{ "location": "New York, US" }}`
```

Figure A6. Prompt template to generate input arguments for a tool

```
Tool name:
{tool_name}
Input details:
{input_desc}
Tool input arguments:
{input_args}
Tool Response:
{tool_response}
---
User Query: {user_query}
---
Provide the response to the user query.
Return the response inside backticks.
Sample response:
`The weather in New York is 30 degrees Fahrenheit and cloudy`
```

Figure A7. Prompt template to generate response for a user query

REFERENCES

[1] G. Dodig-Crnkovic and M. Burgin, “A Systematic Approach to Autonomous Agents,” Philosophies, vol. 9, no. 2, 2024, doi: 10.3390/philosophies9020044.

[2] L. Chourey and L. Prakashchand, “Towards Truly Autonomous AI Agents: Bridging the Gap Between Reactive Responses and Proactive Behaviors,” International Research Journal of Modernization in Engineering, Technology and Science, vol. 6, no. 7, p. 3621, 2024, doi: 10.56726/IRJMETS60636.

[3] D. Berkowitz, “Autonomous AI Agents Are Coming: Why Trust and Training Hold the Keys to Their Success,” Salesforce. [Online]. Available: https://www.salesforce.com/news/stories/ai-training-trust/

[4] P. Garg and D. Beeram, “Large Language Model-Based Autonomous Agents,” International Journal of Computer Trends and Technology, vol. 72, no. 5, pp. 151–162, May 2024, doi: 10.14445/22312803/IJCTT-V72I5P118.

[5] Y. Huang, “Levels of AI Agents: from Rules to Large Language Models,” 2024, arXiv:2405.06643. [Online]. Available: http://arxiv.org/abs/2405.06643

[6] Z. Xi et al., “The Rise and Potential of Large Language Model Based Agents: A Survey,” 2023, arXiv:2309.07864. [Online]. Available: http://arxiv.org/abs/2309.07864

[7] Lareina Yee, Michael Chui, and Roger Roberts, “Why agents are the next frontier of generative AI,” McKinsey. [Online]. Available: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/why-agents-are-the-next-frontier-of-generative-ai

[8] K. R. Haurum, R. Ma, and W. Long, “Real Estate with AI: An agent based on LangChain,” Procedia Computer Science, vol. 242, pp. 1082–1088, Jan. 2024, doi: 10.1016/j.procs.2024.08.199.

[9] Shreepradha Hegde, “A Practical Guide to Building AI Agents With LangGraph,” Association of Data Scientists. [Online]. Available: https://adasci.org/a-practical-guide-to-building-ai-agents-with-langgraph/

[10] Assaf Elovic, “How to Build the Ultimate AI Automation with Multi-Agent Collaboration,” LangChain. [Online]. Available: https://blog.langchain.dev/how-to-build-the-ultimate-ai-automation-with-multi-agent-collaboration/

[11] “Alexa Skills Kit,” Amazon Alexa. [Online]. Available: https://developer.amazon.com/en-US/alexa/alexa-skills-kit

[12] “Alexa Skills Kit - Case Studies,” Amazon Alexa. [Online]. Available: https://developer.amazon.com/en-US/alexa/alexa-skills-kit/get-deeper/case-studies

[13] “Actions on Google,” Google. [Online]. Available: https://developers.google.com/assistant

[14] “Actions console - Google,” Google. [Online]. Available: https://developers.google.com/assistant/console

[15] Nithik Yekollu, Ronit Jain, and Shishir G. Patil, “Agents and Assistants Marketplace,” Agent Marketplace. [Online]. Available: https://gorilla.cs.berkeley.edu/blogs/11_agent_marketplace.html

[16] “Introducing GPTs,” OpenAI. [Online]. Available: https://openai.com/index/introducing-gpts/

[17] “Introducing the GPT Store,” OpenAI. [Online]. Available: https://openai.com/index/introducing-the-gpt-store/

[18] “LangGraph,” LangChain. [Online]. Available: https://www.langchain.com/langgraph

[19] “LangGraph Documentation,” GitHub. [Online]. Available: https://langchain-ai.github.io/langgraph/

[20] AI@Meta, “Llama 3.1 [Language Model].” [Online]. Available: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[21] A. Dubey et al., “The Llama 3 Herd of Models,” 2024, arXiv:2407.21783. [Online]. Available: https://arxiv.org/abs/2407.21783