Generative AI Test Report - Question 2

Nguyen Hoang Gia Khang

***Abstract*—This technical report introduces Large Language Model Agents (LLM agents) and impressive ReAct prompting technique which belong to the Understanding the cognitive architecture of the ReAct Agent section of the technical test. LLM agents' definition and how it works are walked through. Besides, ReAct agents and general agents comparison are done as well.**

***Index Terms*—ReAct, Agent, LLM Agent**

I. Introduction

Although the term "LLM-powered agents" isn't commonly used, it can be defined as a system that can use an LLM to reason through an issue, formulate a solution, and carry out the plan with the aid of a number of tools. Agents are, in essence, systems with the capacity for sophisticated thinking, memory, and task execution. The potential to handle difficult issues with little interaction was initially seen in initiatives like [1] and BabyAGI. Explain like I'm five, For a moment, picture your brain. It is not merely a mass of gray stuff. It's an intricate and intelligent system that processes data, makes choices, and engages with the environment. In a similar vein, the Large Language Model (LLM) is the "brain" of an LLM-based agent. The LLM is the core of an artificial intelligence agent, much as human brain is the center of our existence. Large language models serve as the foundation for AI beings known as LLM Agents, also known as Language Model Agents. Their skill is in comprehending and producing human-like language, which opens up a wide range of uses. Here is the general architecture of an LLM-powered agent application to provide further context for agents. Three pillars define these agents:

- Agent core
- Memory module
- Tools
- Knowledge/Planning Module

A. Agent core

The large language model serves as the cornerstone of the LLM agent. This neural network can produce and comprehend simple sentences because it was trained on enormous datasets. The agent's initial powers and limitations are determined by the size and design of the LLM. In the other words, the key coordinating module that oversees an agent's fundamental reasoning and behavioral traits is known as the agent core. Consider it the agent's "primary decision-making module." Also, this is where we define:

- Overarching objectives of the agent: includes the agent's overarching aims and objectives.
- Implementation instruments: basically a "user manual" or quick summary of all the tools that the agent is able to utilize
- An explanation of how to utilize the various planning modules: information on the benefits of various planning modules and when to apply them.
- Relevant Memory: During inference time, the most pertinent memory fragments from previous exchanges with the user are filled in this dynamic part. It uses the query the user poses to ascertain "relevance."

B. Memory module

The memory of agents is essential because it keeps detailed information pertinent to individual users or tasks and offers a temporal context. Agents use two primary forms of memory to enhance their performance:

- Short-term memory: first, the Large Language Model (LLM) relies heavily on our intrinsic ability to retain short-term memory, which keeps track of current discussions and our most recent actions. The dynamic context window provided by this agent's short-term memory helps it make sense of the interactions it is now engaged in.
- Long-term memory: is the second form of memory in which a large amount of data can be considerably increased by combining the LLM with an external database. With this improvement, the agent's memory is expanded to encompass data, exchanges, and other items from a more extended time frame. By integrating this type of long-term memory, the agent can access accumulated knowledge and insights.

When combined, these bits of data offer the agent a comprehensive understanding of the past as well as contextual knowledge about the user at hand. By taking into account prior contacts, this contextual underpinning gives conversations a more human touch and enhances the agent's dependability and proficiency when doing intricate tasks. In other words, the agent's memory enables them to personalize and engage both sides in intriguing talks, which fosters deeper connections and produces better work outcomes.

C. Tools

Agents can carry out tasks with the help of tools, which are clearly specified executable procedures. They are frequently best understood as specialized third-party APIs. Agents can employ, for example, a code interpreter to solve difficult programming tasks, a RAG pipeline to generate context-aware replies, an API to search the internet for information, or simply a basic API service like an instant messaging application or weather API.

D. Knowledge/Planning module

Knowledge is more generalized competence that may be used for a variety of users and jobs. Information strengthens and expands the base that is already there inside the model's built-in constraints.

One important component that enhances the fundamental design of AI is specialized knowledge. It presents terminologies, concepts, and reasoning styles that are unique to certain subjects or domains. The AI is now able to participate in conversations on these specialist fields more thoroughly and accurately thanks to this augmentation, which increases its value as a resource for users looking for in-depth knowledge in such fields.

Another dimension that enhances the AI's capabilities is commonsense knowledge. Through the introduction of information and insights about science, society, culture, and other fields, it offers a broad perspective of the world that the model would not have. This additional layer of information enables the AI to produce responses that are consistent with human reason, which improves the relatability and realism of interactions.

Procedural knowledge gives the AI the hands-on abilities and techniques needed to complete particular jobs. Procedural knowledge enables the AI to offer useful advice and answers, whether it be for comprehending intricate workflows, utilizing analytical methods, or participating in creative activities.

The AI's ability to understand and participate in meaningful conversations is increased when knowledge is incorporated into its architecture. Even when the AI resets or modifies its memory for a new assignment, knowledge stays relevant. This perfect combination produces AI agents with a plethora of relevant expertise and a reservoir of individualized memories. AI becomes smart, conversational partners that can meet a variety of human needs because to this integration of memory and knowledge.

II. How does LLM Agent work?

After walking through the aforementioned 4 main components of a LLM Agent, we could envision that a LLM Agent is one that can do more than just generate text. It can carry out tasks, have conversations, reason, and even act somewhat autonomously. It does this by using a large language model (LLM) as its central computational engine. In order to control the answers and behaviors of LLM agents, carefully designed prompts are used to encapsulate identities, instructions, permissions, and context. LLM agents have several advantages, one of which is their degree of autonomy. Agents can display self-directed behaviors that vary from being entirely reactive to being quite proactive, depending on the capabilities supplied during the design phase. LLM agents can function semi-autonomously to support humans in a variety of applications, from goal-driven workflow and task automation to conversational chatbots, provided they receive enough prompting and have access to knowledge. Their adaptability and prowess in language modeling open up new avenues for partners in AI that can be customized, comprehend natural language cues, and work alongside human supervision. LLM agents need access to memory, reasoning tools, and knowledge bases in order to become more autonomous. Agents that are trained in prompt engineering possess sophisticated skills in analysis, project planning, execution, retrospective analysis, iterative refinement, and other areas. Agents that possess adequate knowledge and guidance can oversee reasonably autonomous workflows under human supervision. By carefully crafting prompts with identities, instructions, and permissions encoded into them, this ultimately guides the agent's actions. By responding to the AI's output with interactive cues, users effectively direct the agent. Prompts that are well-designed facilitate smooth human-AI interaction.

III. ReAct prompting technique in LLM Agent

Agents can be thought of as LLMs' enabling "tools." Agents enable an LLM to do tasks similar to how a human might use a calculator for math problems or conduct a Google search for information. An LLM can write and run programs with the help of agents. It is capable of doing information searches and SQL database queries.

General agents are similar to "Act-only" which is mentioned in [2] or the illustration is shown in Figure 1. Users' query is seemed to be observation o and an action a is taken based on the context c which is a sequence of previous o and a in ordered. That is the setup of general agents. The pipeline is straightforward which means a problem is solved linearly. There is no any reasoning step on action selection while there are many problems require not only task decomposition but also inference for each step to produce the final response instead of just defaulting to using the tool and returning the results immediately. Chain-of-thoughts prompting technique is got limitation as well. Although it is capable of reasoning, it still uses its parametric memory or internal knowledge which is learned during pretraining to reason an observation it has not been trained yet. The Reason and Act (ReAct) prompting technique is the solution. ReAct prompting technique make an LLM cycle through Reasoning and Action steps. Enabling a multi-step process for identifying answers. The agent uses the ReAct framework to plan the next course of action, formulate a command, and then carry it out. Until the work is finished, the ReAct paradigm iteratively repeats these processes. ReAct is a combination of the Reason Only (Chain-of-thoughts prompting) and Act Only (Self-Ask) paradigms, as Figure 1 illustrates. ReAct is a popular technique because, in addition to giving your LLM the reasoning ahead of time, you also do an action and feed back an observation into the LLM to optimize performance and refine the original logic. In the other words, ReAct prompting technique not only helps LLM reasoning on up-to-date knowledge but also access to external tool. Moreover, iteration strengthens the final result.

IV. Conclusion

In summary, this report introduces Large Language Model Agents (LLM Agents) and how a LLM Agent works. The Reason and Act (ReAct) prompting technique and the differences between general agents and agents applied ReAct are presented. ReAct is a versatile agent that can solve tasks

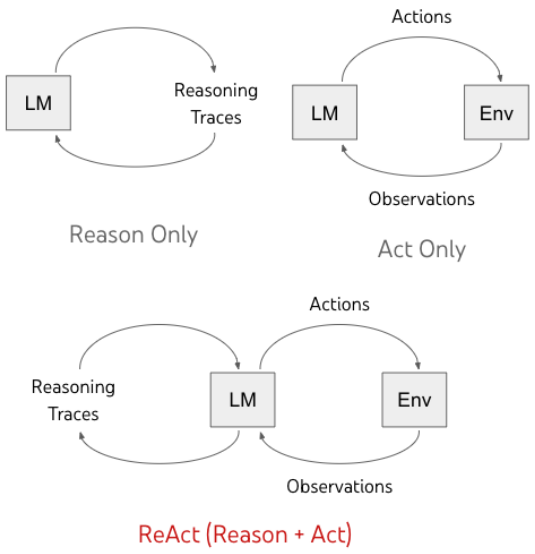

Fig. 1. Source: Previous methods prompt language models (LM) to either generate self-conditioned reasoning traces or task-specific actions. We propose ReAct, a new paradigm that combines reasoning and acting advances in language models.

requiring interactions with the environment by demonstrating the capability of concurrently modeling ideas, actions, and feedback from the environment within a language model.

REFERENCES

[1] H. Yang, S. Yue, and Y. He, "Auto-gpt for online decision making: Benchmarks and additional opinions," 2023.
[2] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," 2023.