Detaillierte Beschreibung der Fähigkeiten eines Agents 251113

Detaillierte Beschreibung der Fähigkeiten eines Agents

Each of the **"tools"** in the system prompt correspond to a function in the code. The agent is going to choose what function to execute and when. Moreover, it is going to decide the parameters that are provided to the functions. The agent is not creating the functions at this point; it is orchestrating their behavior. This means that the logic for how each tool operates is predefined in the code, and the agent focuses on selecting the right tool for the job and providing the correct input to that tool.

Because agents can adapt as the loop progresses, they can dynamically decide which tool to use based on the current context and task requirements. This ability allows the agent to adjust its behavior as new information becomes available, making it more flexible and responsive to the user's input.

For example, if the user asks the agent to read the contents of a specific file, the agent will first use the `list_files` tool to identify the available files. Then, based on the result, it will determine whether to proceed with the `read_file` tool or respond with an error if the file does not exist. The agent evaluates each step iteratively, ensuring its actions are informed by the current state of the environment.

This orchestration process, driven by the agent rules and the tools available, showcases the power of combining pre-defined functions with adaptive decision-making (of the LLM). By allowing the agent to focus on what to do rather than how to do it, we create a system that leverages the LLM for high-level reasoning while relying on well-defined code for execution.

This separation of reasoning and execution is what makes the agent loop so powerful—it creates a modular, extensible framework that can handle increasingly complex tasks without rewriting the underlying tools. Additionally, the agent loop eliminates much of the "glue code"* traditionally required to tie these fundamental functions together. **Instead of hardcoding workflows, the agent dynamically decides the sequence of actions needed to achieve a task, effectively realizing a program on top of its components.** This dynamic nature enables the agent to combine its tools in ways that would typically require custom logic, making it far more versatile and capable of addressing a broader range of use cases **without additional development overhead**.

*Der Agent Loop ersetzt oder automatisiert viel von dem "glue code" Kleber-Code, den man früher manuell schreiben musste, um die einzelnen Funktionen (wie Denken, Handeln, Gedächtnis) zu verbinden.

Detaillierte Beschreibung der Fähigkeiten eines Agents

Jedes der „Werkzeuge" in der Systemaufforderung entspricht einer Funktion im Code. Der Agent entscheidet, welche Funktion wann ausgeführt wird. Darüber hinaus legt er die Parameter fest, die den Funktionen zur Verfügung gestellt werden.

Der Agent erstellt zu diesem Zeitpunkt keine Funktionen, sondern koordiniert deren Verhalten. Das bedeutet, dass die Logik für die Funktionsweise jedes Tools im Code vordefiniert ist und sich der Agent darauf konzentriert, das richtige Tool für die Aufgabe auszuwählen und die richtigen Eingaben für dieses Tool bereitzustellen.

Da sich Agenten im Verlauf der Schleife anpassen können, können sie dynamisch entscheiden, welches Tool sie basierend auf dem aktuellen Kontext und den Aufgabenanforderungen verwenden. Diese Fähigkeit ermöglicht es dem Agenten, sein Verhalten anzupassen, sobald neue Informationen verfügbar werden, wodurch er flexibler wird und besser auf die Eingaben des Benutzers reagieren kann.

Wenn der Benutzer den Agenten beispielsweise auffordert, den Inhalt einer bestimmten Datei zu lesen, identifiziert der Agent zunächst mit dem Tool „list_files" die verfügbaren Dateien. Anhand des Ergebnisses entscheidet er dann, ob er mit dem Tool „read_file" fortfährt oder eine Fehlermeldung ausgibt, wenn die Datei nicht vorhanden ist. Der Agent bewertet jeden Schritt iterativ und stellt so sicher, dass seine Aktionen auf dem aktuellen Zustand der Umgebung basieren.

Dieser Orchestrierungsprozess, der von den Agentenregeln und den verfügbaren Tools gesteuert wird, zeigt die Leistungsfähigkeit der Kombination vordefinierter Funktionen mit adaptiver Entscheidungsfindung (des LLM). Indem wir dem Agenten ermöglichen, sich auf das Was statt auf das Wie zu konzentrieren, schaffen wir ein System, das das LLM für hochgradiges Denken nutzt und sich gleichzeitig auf klar definierten Code für die Ausführung stützt.
Diese Trennung von Schlussfolgerung und Ausführung macht die Agentenschleife so leistungsstark – sie schafft ein modulares, erweiterbares Framework, das immer komplexere Aufgaben bewältigen kann, ohne die zugrunde liegenden Tools neu schreiben zu müssen.
Darüber hinaus eliminiert die Agentenschleife einen Großteil des „Glue-Codes"*, der traditionell erforderlich ist, um diese grundlegenden Funktionen miteinander zu verbinden. Anstatt Workflows fest zu programmieren, entscheidet der Agent dynamisch über die Abfolge der Aktionen, die zur Erfüllung einer Aufgabe erforderlich sind, und realisiert so effektiv ein Programm auf der Grundlage seiner Komponenten. Diese Dynamik ermöglicht es dem Agenten, seine Tools auf eine Weise zu kombinieren, die normalerweise eine benutzerdefinierte Logik erfordern würde, wodurch er weitaus vielseitiger wird und ohne zusätzlichen Entwicklungsaufwand ein breiteres Spektrum von Anwendungsfällen abdecken kann.

*Der Agent Loop ersetzt oder automatisiert viel von dem "glue code" Kleber-Code, den man früher manuell schreiben musste, um die einzelnen Funktionen (wie Denken, Handeln, Gedächtnis) zu verbinden.