LLM Agentic Systems

Calvin Chi

2025-07-18

Table of contents

Preface

In March of 2025, I had the new exciting opportunity to join a new team developing foundational models and AI agents. Since the field of agents is new and rapidly evolving, the number of established texts or formal courses on AI agents is sparse. A common joke in this field is that by the time someone writes a book, half of the content becomes outdated when finished. Instead, most people that I know stay up to date with the latest in the field by continually reading blogs, tweets, and papers. However, without a comprehensive text, it can be hard to see how different concepts fit together within the bigger picture. This is true especially when definitions are still being debated, new perspectives are still being developed, and new terms are being invented to offer greater conceptual clarity. A recent example is the term "context engineering", which gained popularity after a discussion on X on June of 2025, and was introduced to unify the goals of prompt engineering, history management, tool use, etc.

I started this online book as an attempt to organize and aggregate the lessons I learned along the way as I build agents. The structure of a book is helpful as it better organizes concepts and ideas. The book is meant to be online to easily accomodate updates in the field, which are frequent. The contents are drawn from my learnings from research papers, blogs, talks, and the practical experience of building agents. My goal is to write in sufficient levels of depth and detail such that it becomes clear how things work "underneath the hood". For example, mechansims are explicitly described and illustrated concretely with code when helpful. It is my belief that it is this level of detail and concreteness that is going to be most helpful for people looking to jump in and start building. Given how new and evolving the field of AI agents are, the definitions and perspectives of this book may not necessarily align with everyone's definitions and perspectives, and may not even stand the test of time. Nor is it meant to be comprehensive. However, that is okay as long as it proides *one* valid mental model and helps readers get the bigger picture in order to get started with building agents.

Part I

Concepts

1 Introduction

In artifical intelligence (AI), an agent is broadly defined as anything that can perceive and act in its own environment (Norvig and Intelligence 2002). With the rise of large language models (LLMs), LLMs are now used to power modern agentic systems by leveraging the general intelligence capabilities of a LLM (Brown et al. 2020). At its best, a LLM can dynamically decide the sequence of steps that need to be executed in order to accomplish a given task, essentially achieving autonomy.

In practice, agentic systems differ in the degree of reliance on the LLM as a decision maker, since the increased flexibility that LLMs provide also comes at the cost of reliability. On one end of the spectrum is a LLM workflow, which has LLMs participate in a limited scope within a broader predefined workflow. On the other end of the spectrum is a LLM agent, where the LLM directs its own workflow to accomplish a task. We can illustrate the difference between a workflow and an agent with a customer service chatbot example:

- **Workflow**: a potential workflow executes (1) intent classification by a LLM, (2) tool execution based on intent, and (3) LLM response generation. Based on the determined intent, only one tool is executed by following a pre-defined if-else control flow.
- **Agent**: given a set of tools, a LLM dynamically decides which tool to use in response to customer inquiry. In this process, multiple tools can be used any number of times, with the steps planned or decided by the LLM itself. Once the LLM determines it has collected sufficient information from tool-use to respond, it generates a final response to the customer.

While an agent can tackle tasks more adaptively, it is also less predictable and reliable. On the other hand, workflows are more deterministic and thus more reliable, but they are limited in their ability to tackle more open-ended tasks where there may not be one obvious approach. Choosing between a workflow and an agent requires considering the balance of flexibility and reliability needed for the application. In the rest of this book, the word *agent* will be used interchangeably with agentic systems, with the distinction between workflows and agents expliclty called out only when necessary.

Building an agentic system from a LLM requires a prompt, tools, and memory. The prompt is piece of text that instructs the LLM on how to behave within the agent application. Tools allow an agent to take actions and is typically assessed by an agentic system in the form of an API. Finally, memory allows an agent to act and behave in a contextualized manner, with user information or conversation history being common memory contexts. Each of these components are the building blocks that can be used to create and shape a LLM agent. Figure 1.1 illustrates an agentic system and its components:

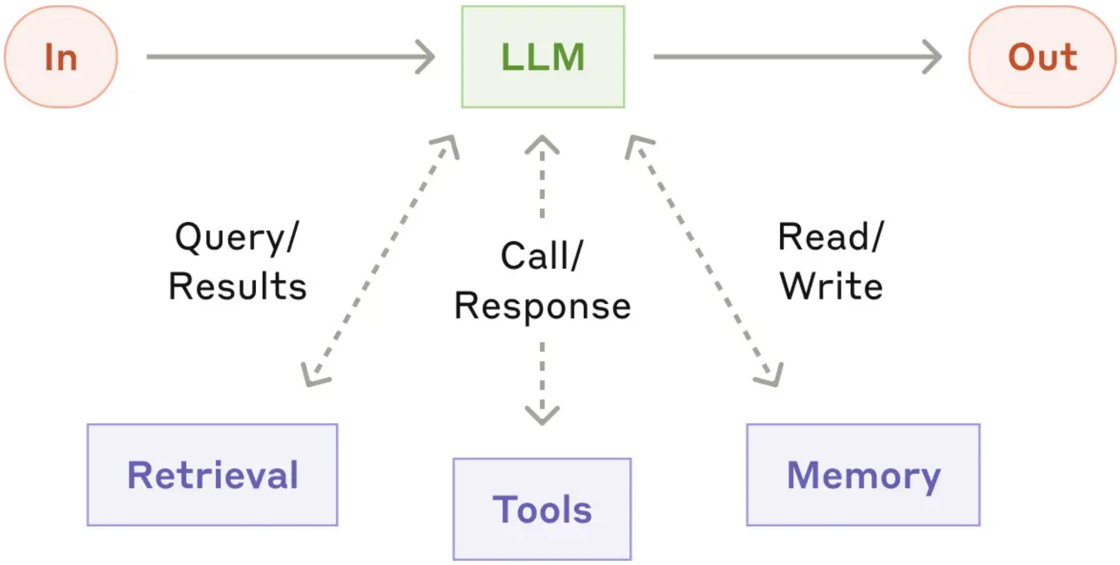

Figure 1.1: An agentic system and its components. Dotted and double-sided arrows indicate that the interaction is optional and bidirectional respectively. Additionally, the double-sideness implies that the interaction can be iterative, occuring multiple times until the LLM determines its task is done. Source: "Building Effective Agents" (https://www.anthropic.com/index/building-effective-agents).

1.1 Prompt

A prompt is a piece of text that instructs a LLM how to behave within an agent application. Conceptually, a prompt can be organized into a system prompt, contextual prompt, role prompt, and a user prompt. In the end, they are all concatenated into a single text input when invoking the LLM (i.e. asking LLM to generate repsonse).

- **System prompt**: contains high level instructions

1.2 Tools

1.3 Memory

```python
def tool(city):
    if city == "New York":
        return "The weather in New York is sunny."
    elif city == "Los Angeles":
        return "The weather in Los Angeles is warm."
    else:
        return "Weather information for this city is not available."
```

2 Summary

In summary, this book has no content whatsoever.

References

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. "Language Models Are Few-Shot Learners." *Advances in Neural Information Processing Systems* 33: 1877–1901.

Norvig, P Russel, and S Artificial Intelligence. 2002. "A Modern Approach." *Prentice Hall Upper Saddle River, NJ, USA: Rani, M., Nayak, R., & Vyas, OP (2015). An Ontology-Based Adaptive Personalized e-Learning System, Assisted by Software Agents on Cloud Storage. Knowledge-Based Systems* 90: 33–48.