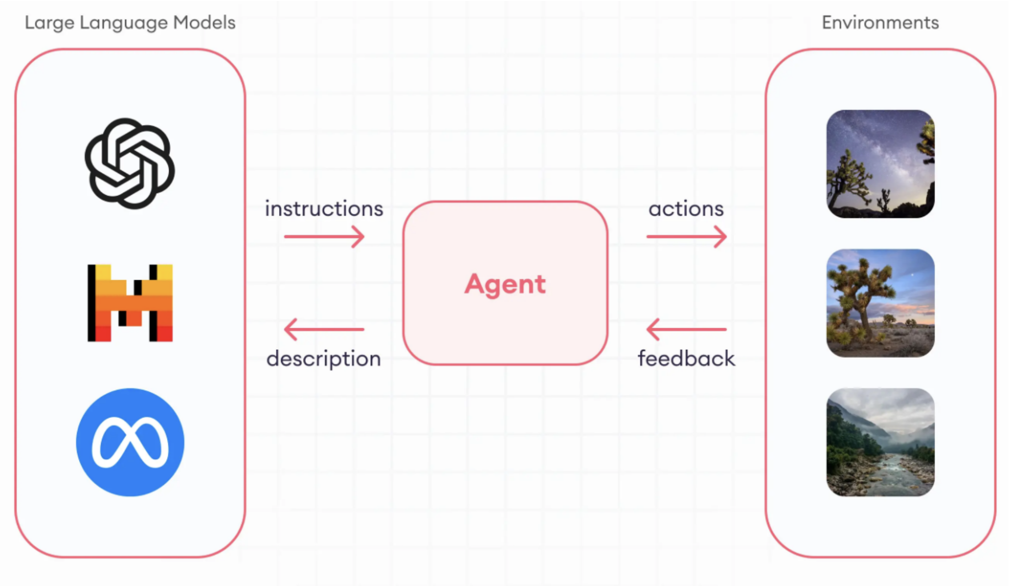
Alberto Andrés Valdés González.
Degree: Mathematical Engineer.
Work position: ML-Engineer.
Mail: anvaldes@uc.cl/alberto.valdes.gonzalez.96@gmail.com
Location: Santiago, Chile.

LLM Agents



What are LLM agents?

LLM agents are advanced AI systems designed for creating complex text that needs sequential reasoning. They can think ahead, remember past conversations, and use different tools to adjust their responses based on the situation and style needed.

Consider a question in the legal field that sounds like this:

"What are the potential legal outcomes of a specific type of contract breach in California?"

A basic LLM with a retrieval augmented generation (RAG) system can easily fetch the needed information from legal databases.

Now, consider a more detailed scenario:

Ïn light of new data privacy laws, what are the common legal challenges companies face, and how have courts addressed these issues?"

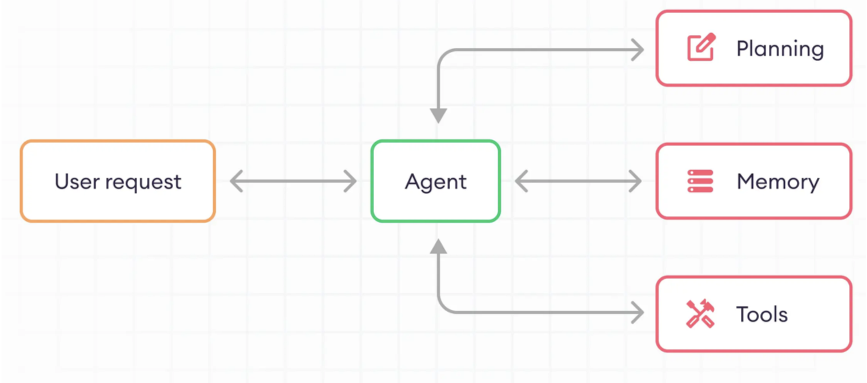
This question digs deeper than just looking up facts. It's about understanding new rules, how they affect different companies, and finding out what courts have said about it all. A simple RAG system can pull up relevant laws and cases, but it lacks the ability to connect these laws to actual business situations or analyze court decisions in depth.

In such situations, LLM agents step in. When the project demands sequential reasoning, planning, and memory, LLM agents shine.

For this question, the agent can break down its tasks into subtasks like so. The first subtask may be accessing legal databases to retrieve the latest laws and regulations. Secondly, it can establish a historical baseline of how similar issues were previously handled. Another subtask can be summarizing legal documents and forecasting future trends based on observed patterns.

To complete these subtasks, the LLM agent requires a structured plan, a reliable memory to track progress, and access to necessary tools. These components form the backbone of an LLM agent's workflow.

LLM agent components



Agent/Brain:

At the core of an LLM agent is a language model that processes and understands language based on a vast amount of data it's been trained on.

When you use an LLM agent, you start by giving it a specific prompt. This prompt is crucial—it guides the agent on how to respond, what tools to use, and the goals it should aim to achieve during the interaction. It's like giving directions to a navigator before a journey.

Additionally, you can customize the agent with a specific persona. This means setting up the agent with certain characteristics and expertise that make it better suited for particular tasks or interactions. It's about tuning the agent to perform tasks in a way that feels right for the situation.

Essentially, the core of an LLM agent combines advanced processing abilities with customizable features to effectively handle and adapt to various tasks and interactions.

Memory:

The memory of LLM agents helps them handle complex LLM tasks with a record of what's been done before. There are two main memory types:

Short-term memory: This is like the agent's notepad, where it quickly writes down important details during a conversation. It keeps track of the ongoing discussion, helping the model respond relevantly to the immediate context. However, this memory is temporary, clearing out once the task at hand is completed.

Long-term memory: Think of this as the agent's diary, storing insights and information from past interactions over weeks or even months. This isn't just about holding data; it's about understanding patterns, learning from previous tasks, and recalling this information to make better decisions in future interactions.

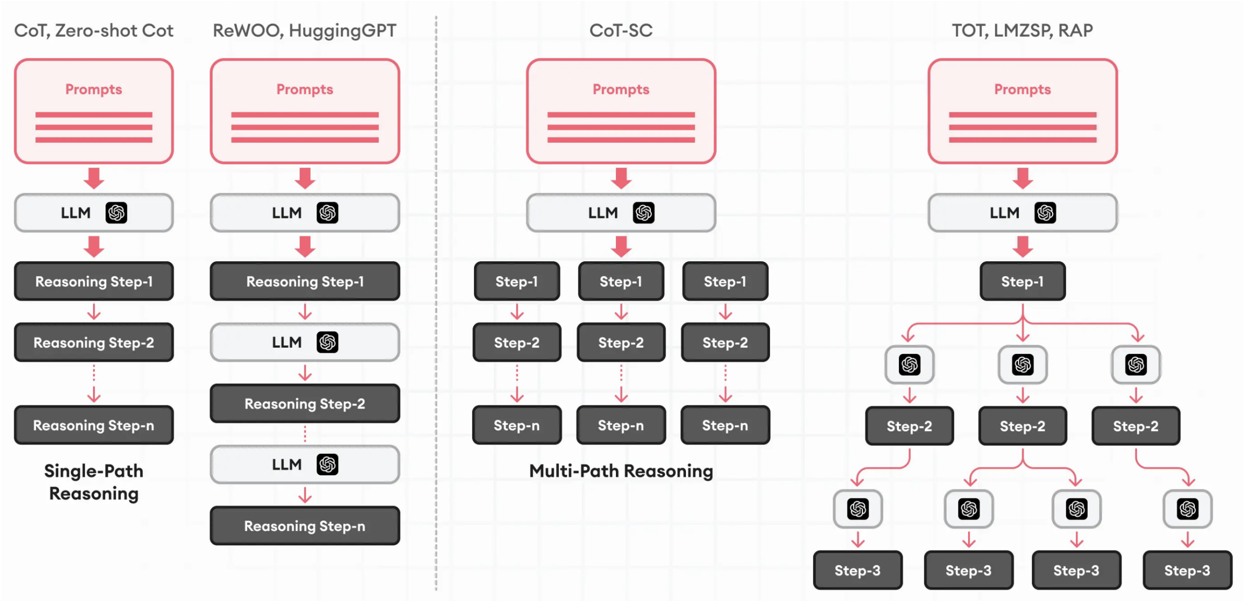
By blending these two types of memory, the model can keep up with current conversations and tap into a rich history of interactions. This means it can offer more tailored responses and remember user preferences over time, making each conversation feel more connected and relevant. In essence, the agent is building an understanding that helps it serve you better in each interaction.

Planning;

Through planning, LLM agents can reason, break down complicated tasks into smaller, more manageable parts, and develop specific plans for each part. As tasks evolve, agents can also reflect on and adjust their plans, making sure they stay relevant to real-world situations. This adaptability is key to successfully completing tasks.

Planning typically involves two main stages: plan formulation and plan reflection.

Plan formulation: During this stage, agents break down a large task into smaller sub-tasks. Some task decomposition approaches suggest creating a detailed plan all at once and then following it step by step. Others, like the chain of thought (CoT) method, recommend a more adaptive strategy where agents tackle sub-tasks one by one, allowing for greater flexibility. Tree of thought (ToT) is another approach that takes the CoT technique further by exploring different paths to solve a problem. It breaks the problem into several steps, generating multiple ideas at each step and arranging them like branches on a tree.



There are also methods that use a hierarchical approach or structure plans like a decision tree, considering all possible options before finalizing a plan. While LLM-based agents are generally knowledgeable, they sometimes struggle with tasks that require specialized knowledge. Integrating these agents with domain-specific planners has proven to improve their performance.

Plan reflection: After creating a plan, it's important for agents to review and assess its effectiveness. LLM-based agents use internal feedback mechanisms, drawing on existing models to refine their strategies. They also interact with humans to adjust their plans based on human feedback and preferences. Agents can also gather insights from their environments, both real and virtual, using outcomes and observations to refine their plans further.

Two effective methods for incorporating feedback in planning are ReAct and Reflexion.

ReAct, for instance, helps an LLM solve complex tasks by cycling through a sequence of thought, action, and observation, repeating these steps as needed. It takes in feedback from the environment, which can include observations as well as input from humans or other models. This method allows the LLM to adjust its approach based on real-time feedback, enhancing its ability to answer questions more effectively.

Tools use:

Tools in this term are various resources that help LLM agents connect with external environments to perform certain tasks. These tasks might include extracting information from databases, querying, coding, and anything else the agent needs to function. When an LLM agent uses these tools, it follows specific workflows to carry out tasks, gather observations, or collect the information needed to complete subtasks and fulfill user requests.