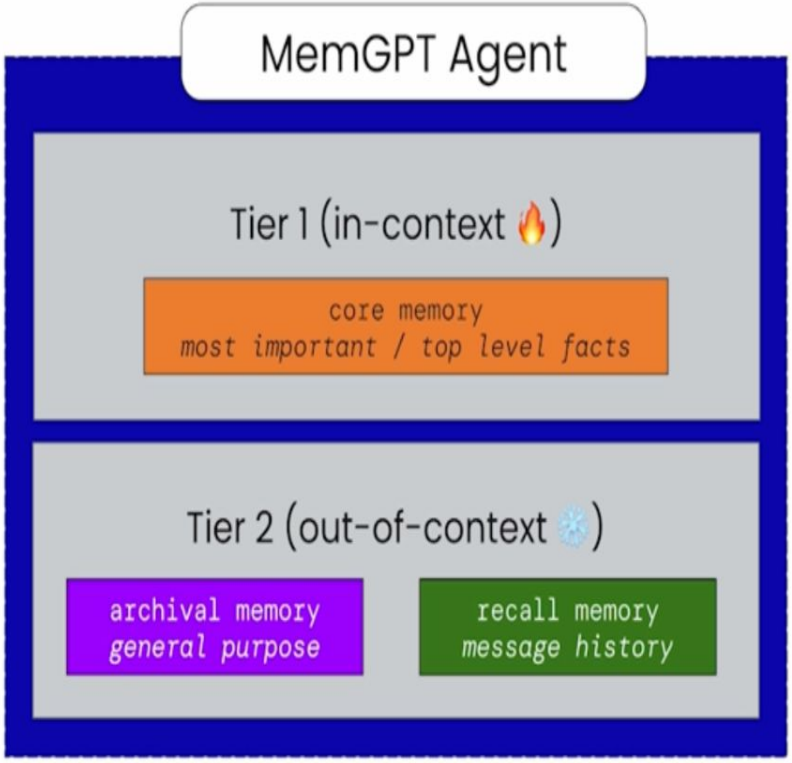
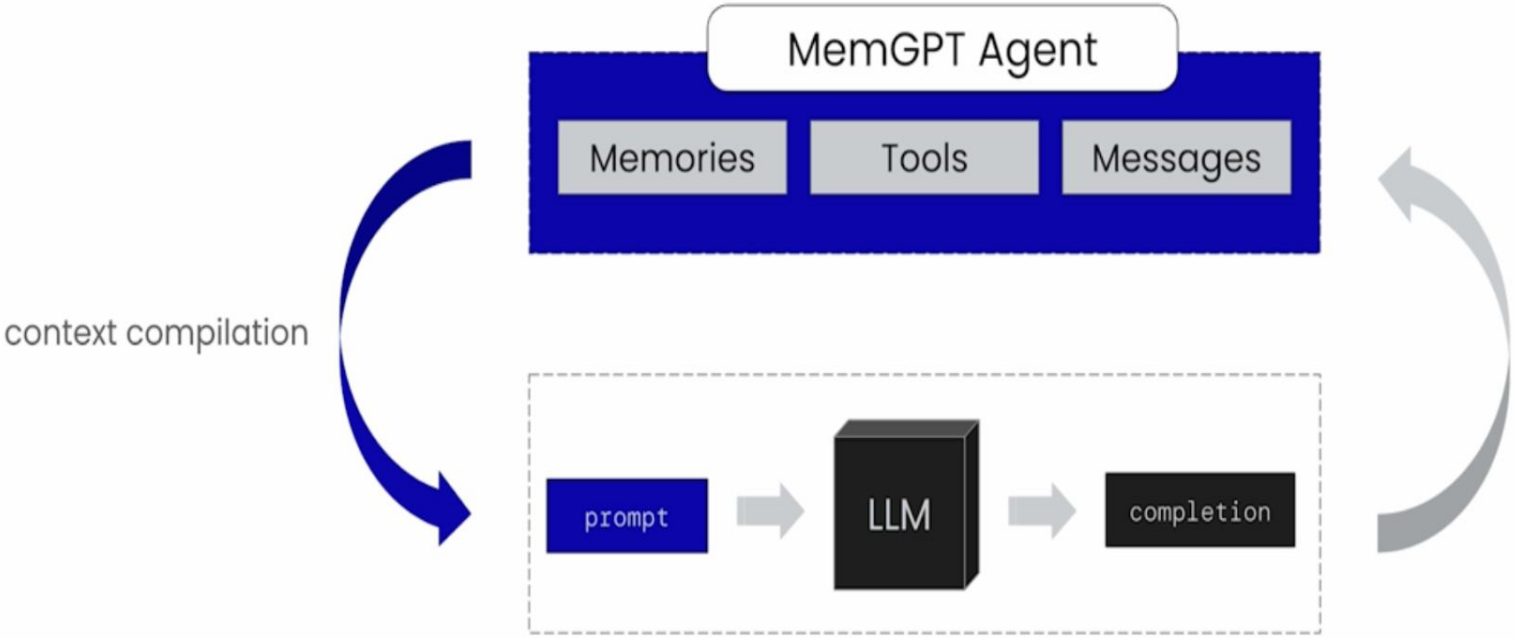
MemGPT

MemGPT agents have two tiers of memory - **core memory** (tier 1) and **archival/recall** (tier 2)



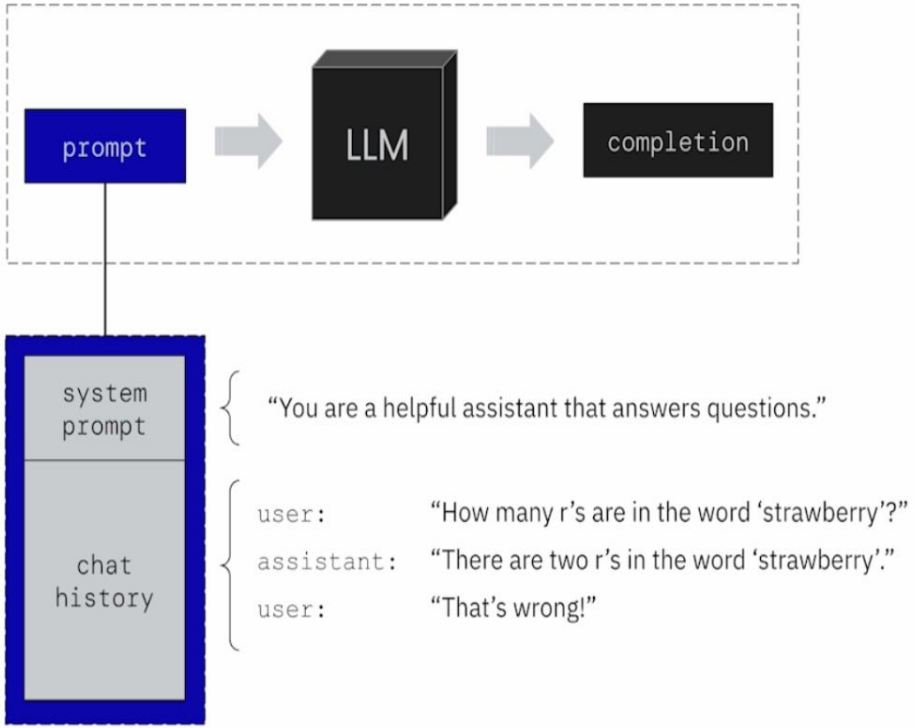
Breaking down the context window

How can we turn our **agent state** into a **prompt**?



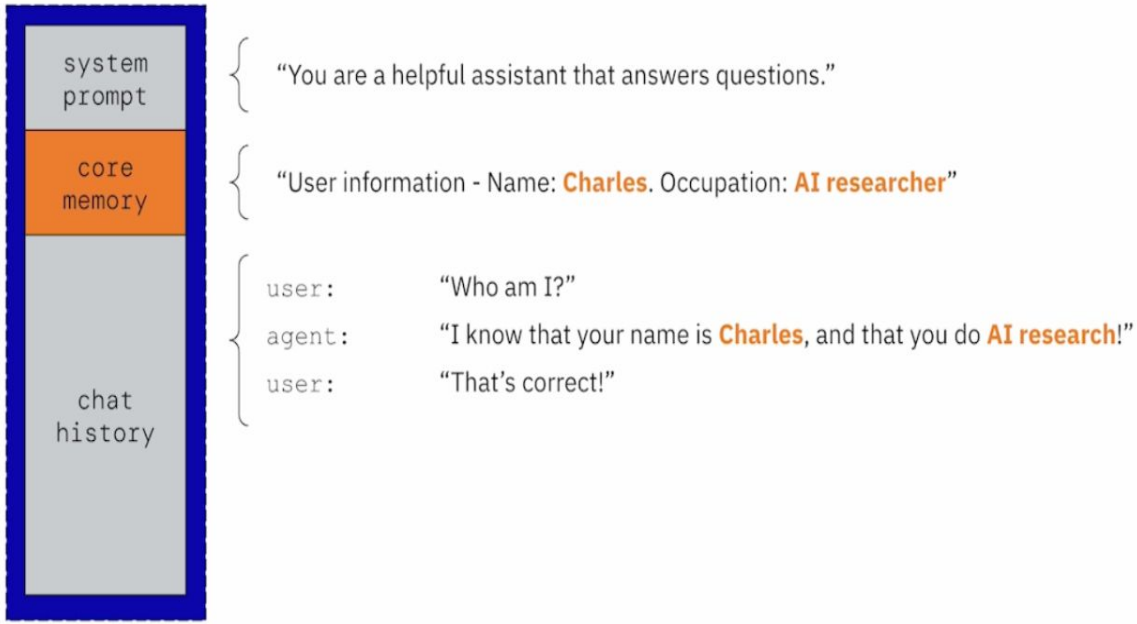
Breaking down the context window

In most popular LLM APIs ("Chat Completion"),
the LLM input is divided into "**system prompt**" and "**chat history**"



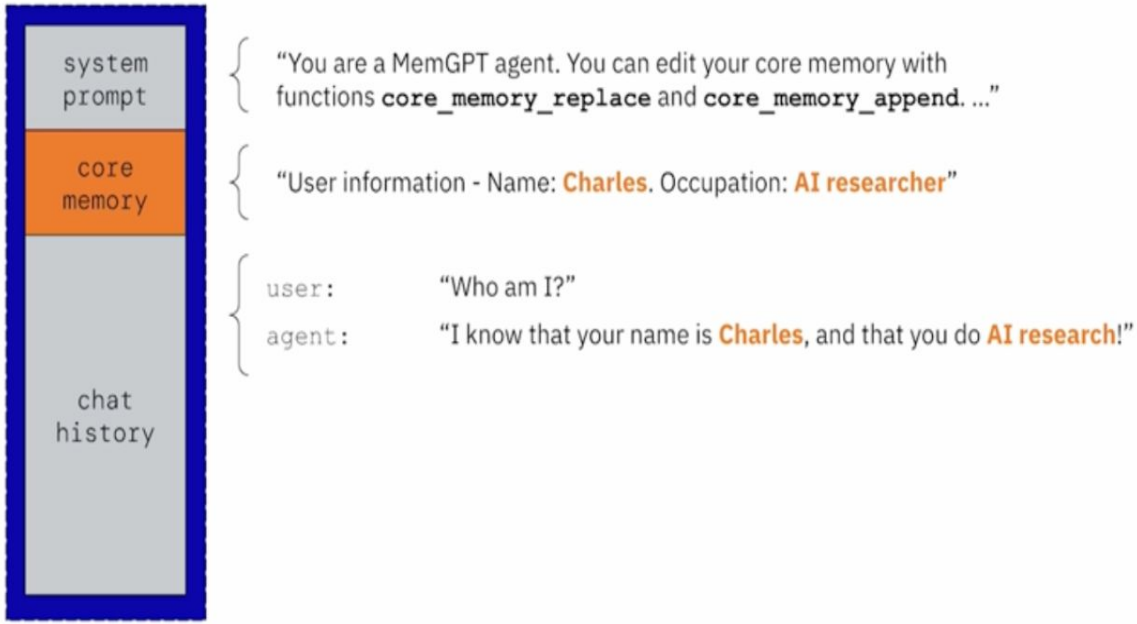
Breaking down the context window

In a MemGPT agent, we add reserve a piece of the context window for **core memory**



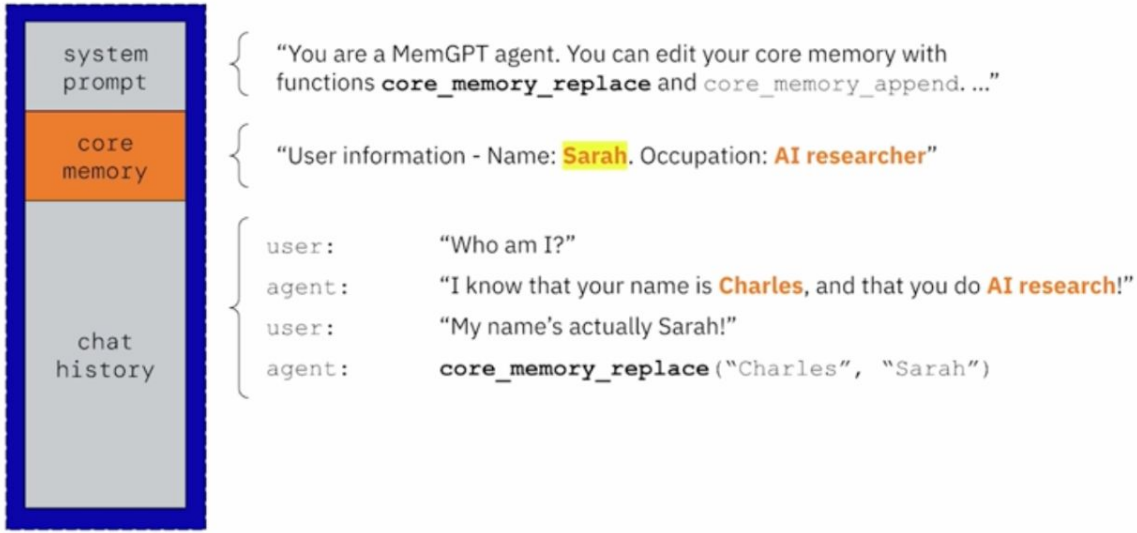
Breaking down the context window

The system prompt includes information about **how to edit** core memory



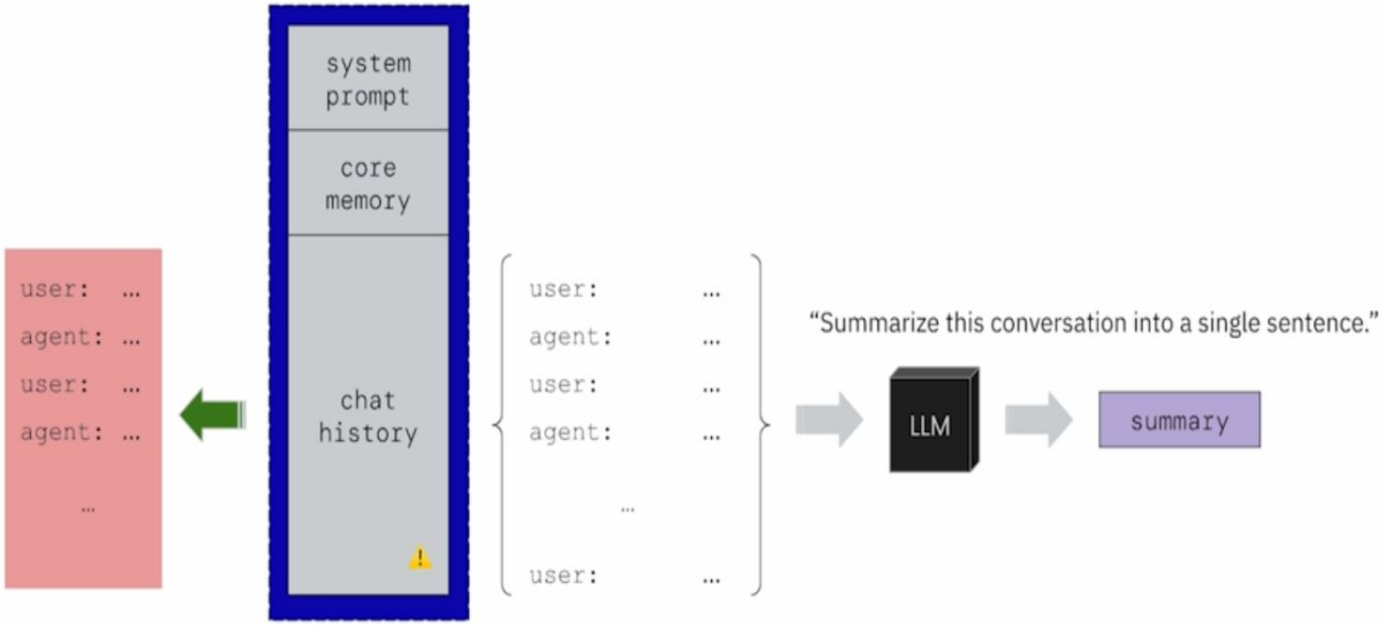
Breaking down the context window

The system prompt includes information about **how to edit** core memory



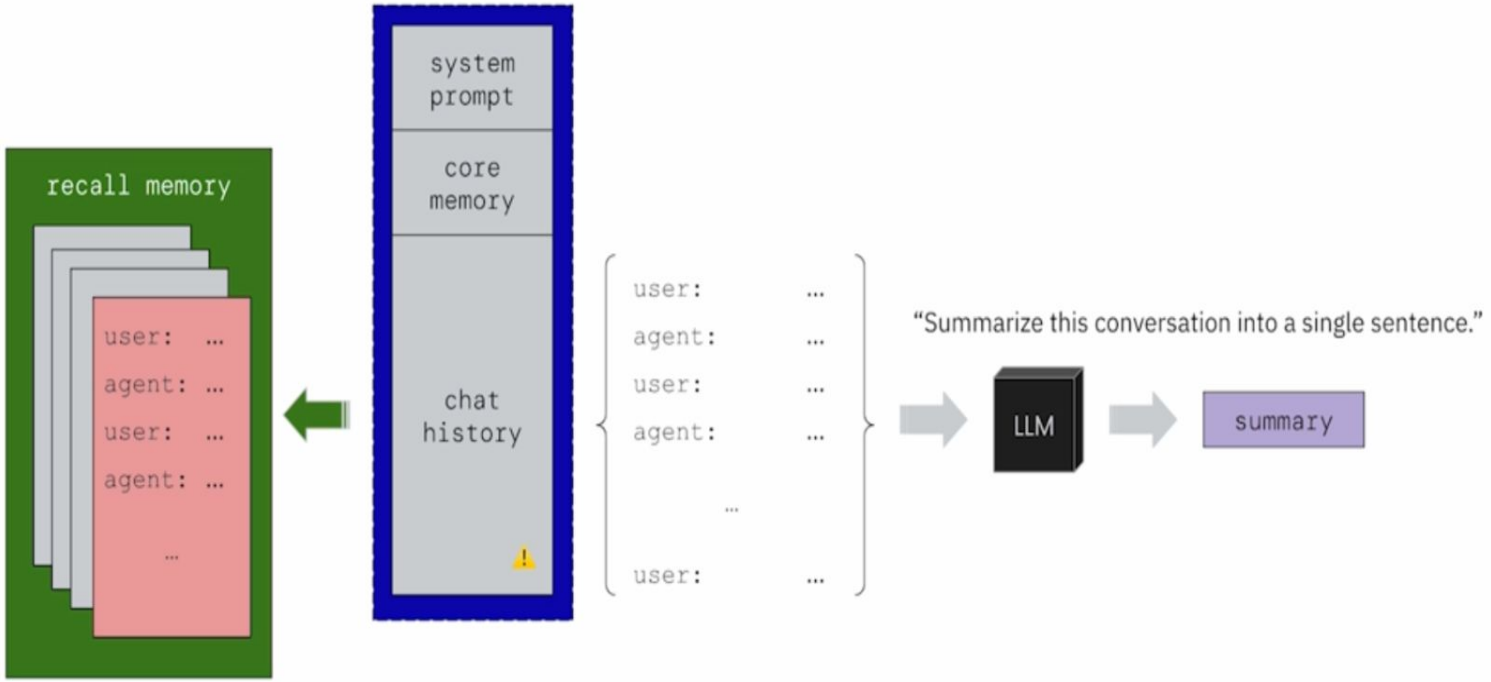
Breaking down the context window

When we run out of space: **summarize** and **flush** (evict) memory



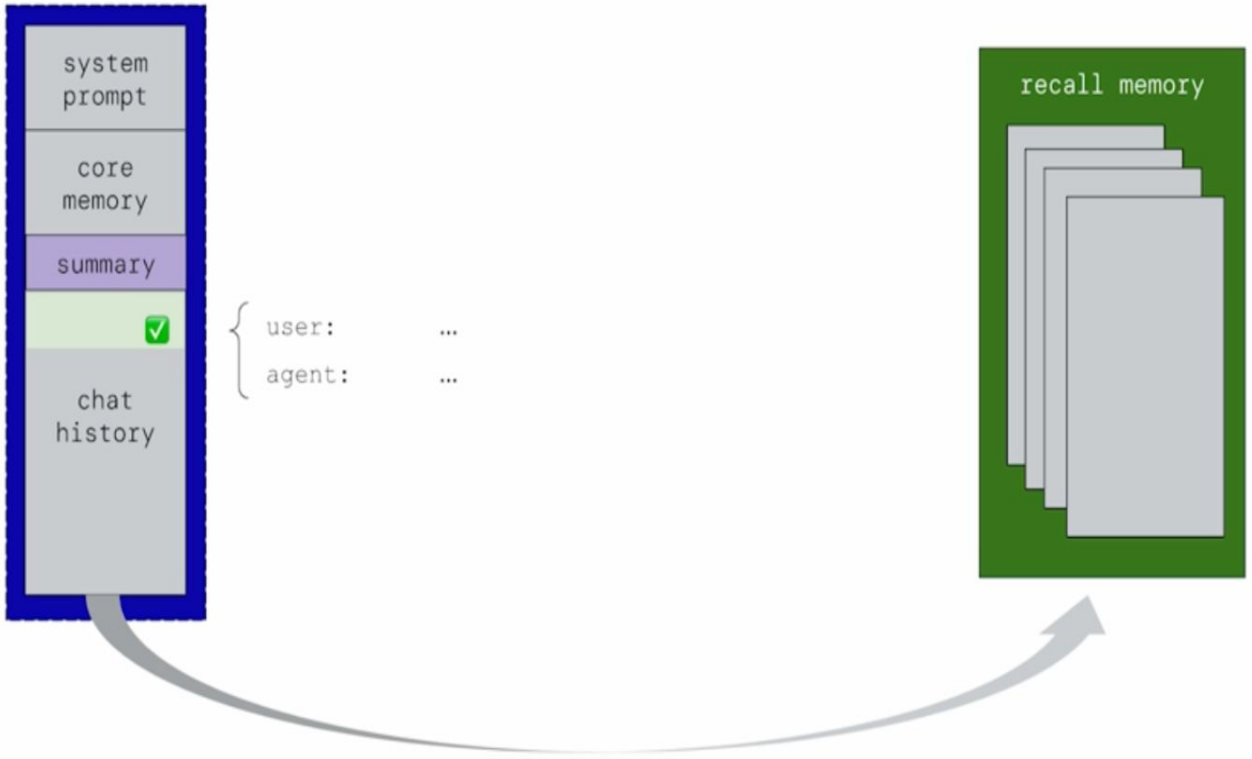
Breaking down the context window

We store the evicted messages inside on "disk" (an "unlimited" data store)



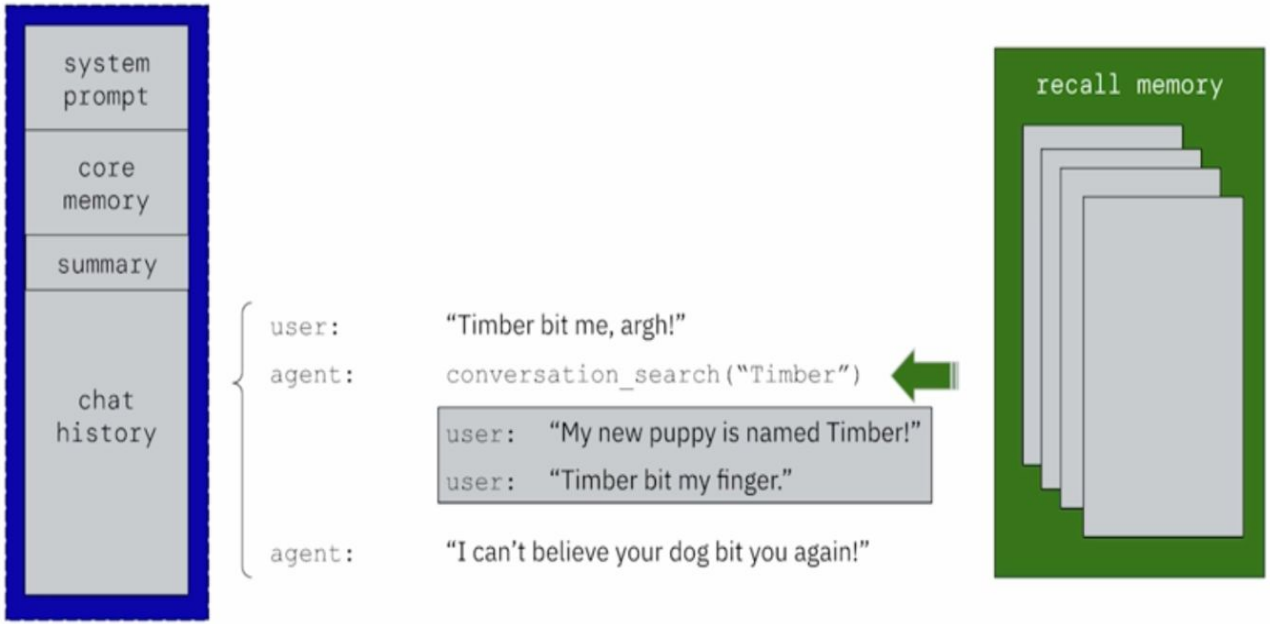
Breaking down the context window

By flushing messages to external storage, we "reset" our context window



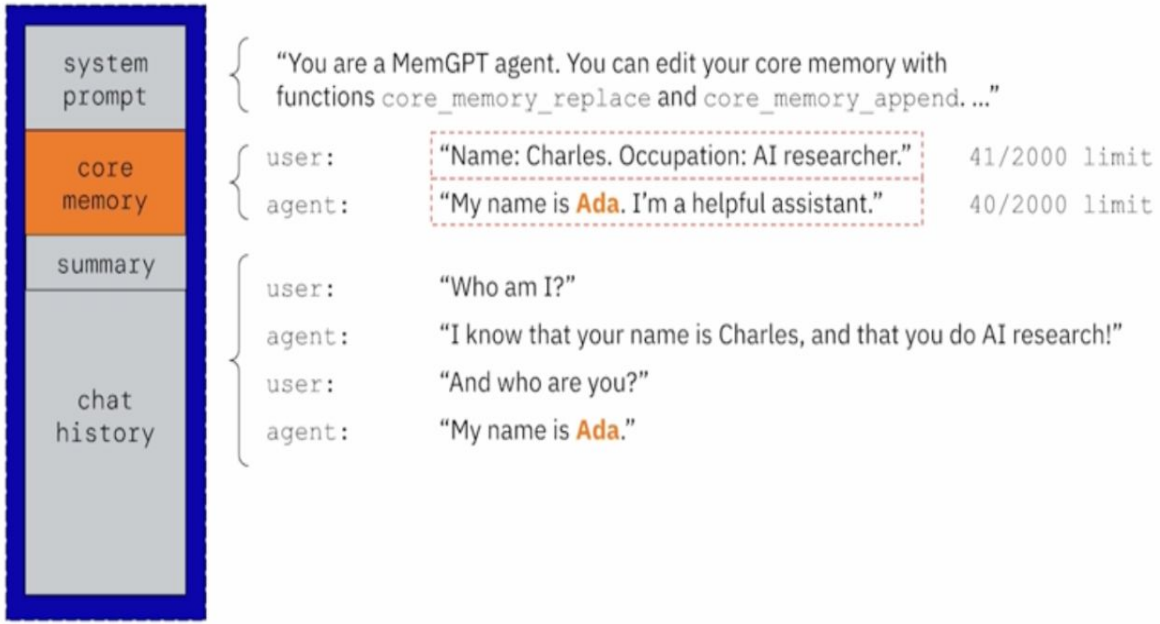
Breaking down the context window

If the agent needs access to old messages, it can **search** recall storage



Breaking down the context window

Core memory is also limited in size



Breaking down the context window

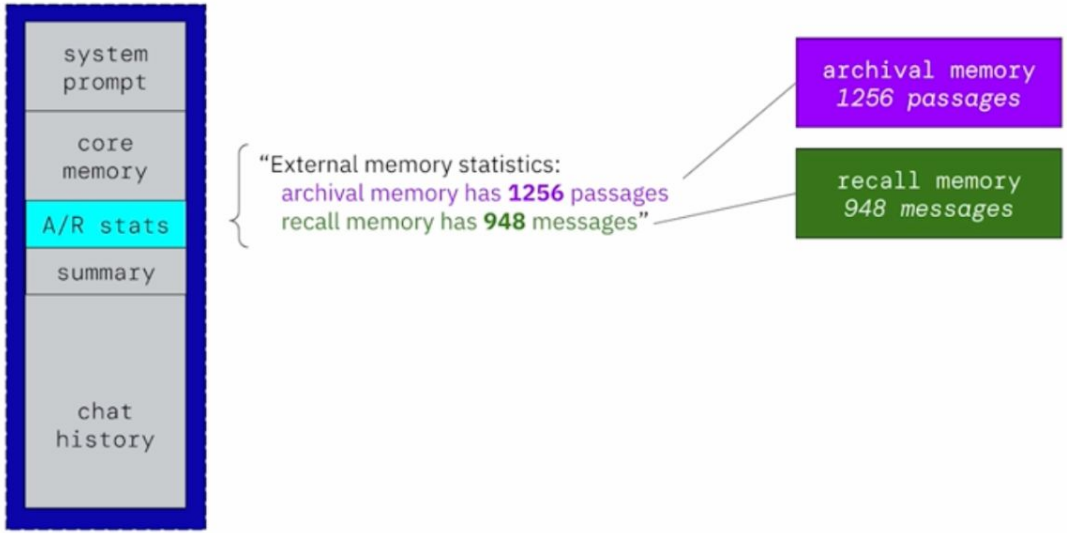
MemGPT agents have a "second tier" of memory: **archival memory**

Breaking down the context window

MemGPT agents have a "second tier" of memory: **archival memory**

Breaking down the context window

External memory statistics help the MemGPT agent "know what it doesn't know"



Breaking down the context window

External memory statistics help the MemGPT agent "know what it doesn't know"