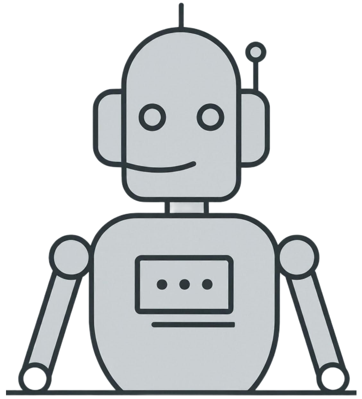
Tool Use



AI = Intelligent computer programs that can perceive, reason, learn, and act in complex environments (Russell & Norvig, 2022)

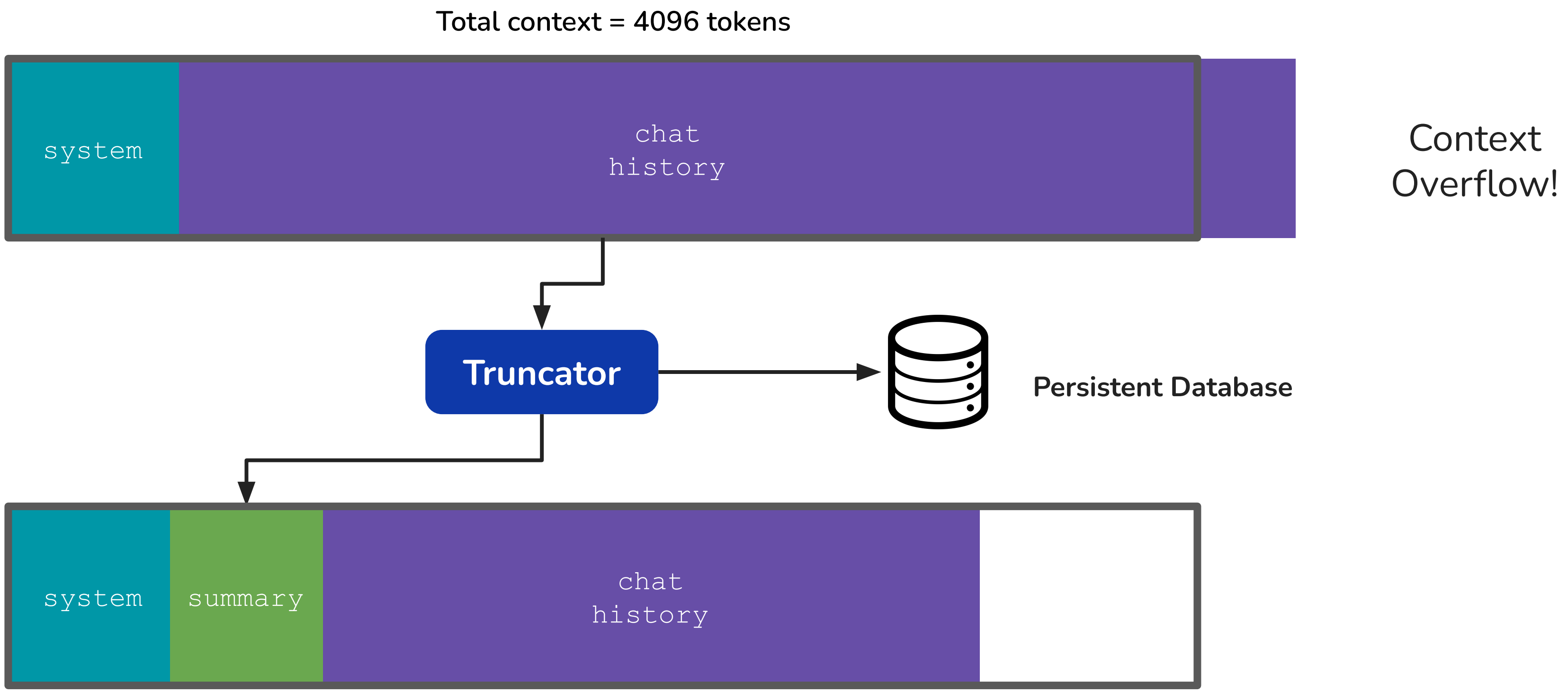
Agenda

In this session, we will discuss:

- Handling Conversational Agent Memory with Tools
- Wrapping Functions as LLM Tools

Managing LLM Context

Actively Managing Conversation Memory

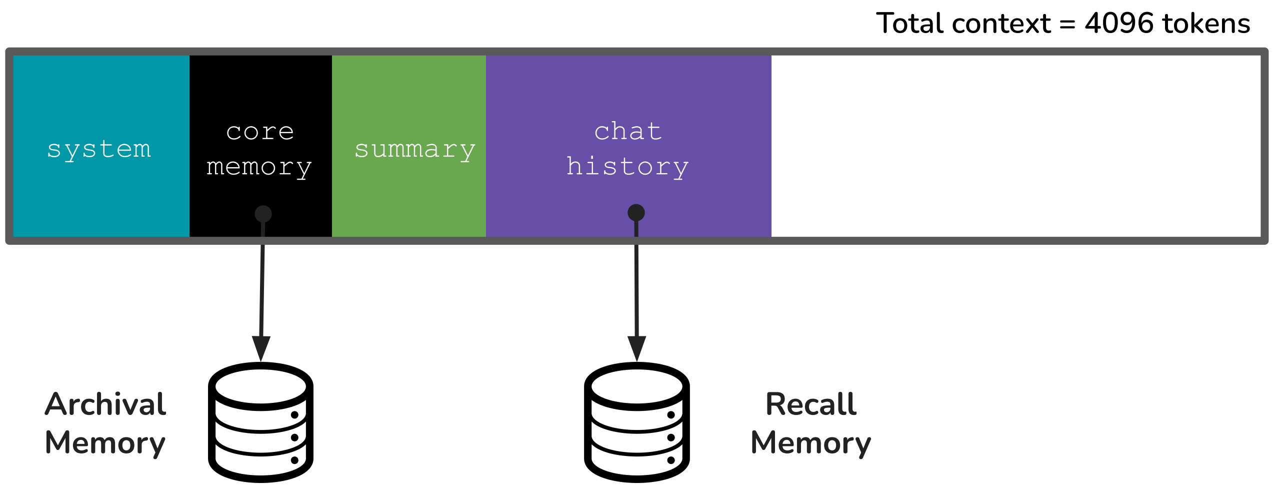


Chat history can be periodically summarized/truncated and appended to the system message while chat history is flushed to a persistent store.

Actively Managing Conversation Memory

Note:
Agentic patterns often require long conversation threads. Thai poses an important challenge: the chat history required to execute the objective becomes larger than that available for the conversation. Even when a model has a large context window, it becomes difficult for the model to navigate this quickly to provide appropriate answers. One way to manage the context window is to implement a Truncator that periodically flushes the chat history to persistent datastore (say an SQL or NoSQL database) and injects a summary of this history back to the context. Truncators implement a controllable way to manage memory.

Managing Session Memory - MemGPT



Core memory, i.e., information that is needed for the current session is retrieved from archival memory and frozen during the conversation. Updates to core memory post session are indexed to archival memory. Overflows in chat history are stored in a searchable recall memory. LLMs decide which information from a current session should be retained in core memory.

Managing Session Memory - MemGPT

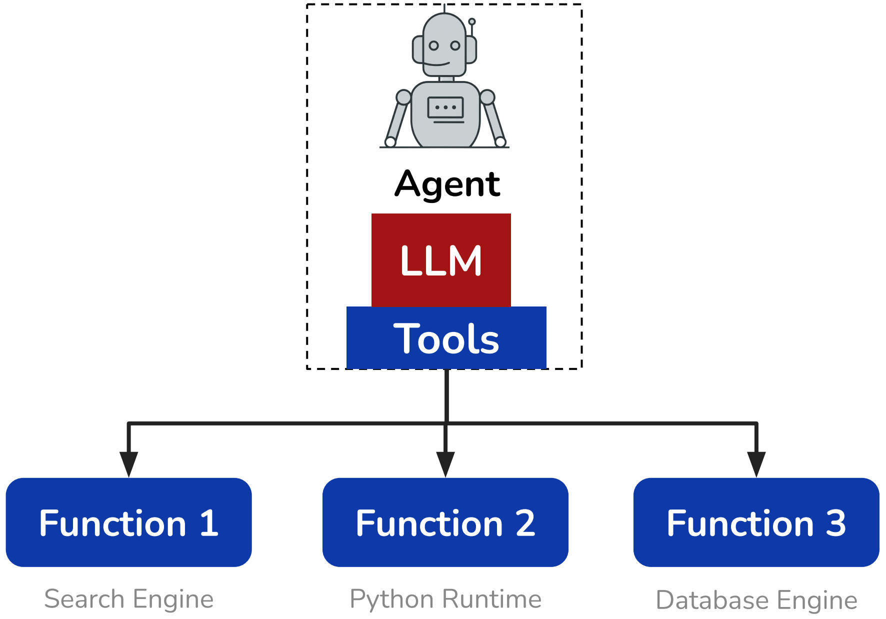
Note:
The MemGPT paper introduced another important way to organize the context sent to the LLM in a more meaningful way. In this paradigm, memory is divided into core memory, that is, information critical for the current conversation (e.g., customer account details, preferences) are stored in archival memory and retrieved before the chat session begins. Once the session concludes, the LLM decides if the archival memory for the customer needs to be updated. Chat history is periodically flushed to a searchable recall memory and a summary is retained in the context.

Hands-on

Implementing active memory management

Function Calling

Function Calling



Any Python function can be converted to a tool and bound to an LLM. When a user query is received, the LLM chooses the appropriate function to call and generates appropriate function arguments. Compound queries are handled by a sequence of function calls executed one after another.

API calls to other models could also be wrapped as functions

Function calling agents are defined by the functions they can execute; they are a great way to build deterministic agents

Hands-on

Implementing Function Calling Agents

Great Learning
POWER AHEAD

Thank You