arxiv.org/abs/2507.21428

MemTool: Optimizing Short-Term Memory Management for Dynamic Tool Calling in LLM Agent Multi-Turn Conversations

MemTool：在多轮对话中如何动态管理每一轮query的工具调用集合

Elias Lumer, Anmol Gulati, Vamse Kumar Subbiah,
Pradeep Honaganahalli Basavaraju and James A. Burke
Commercial Technology and Innovation Office, PricewaterhouseCoopers, U.S.A

未开源

简介

本文提出MemTool，目的是让agent在多轮对话中能够动态管理针对每一轮query所需要的最合适的工具集合。MemTool内置了三种模式：Autonomous Agent Mode、Workflow Mode和Hybrid Mode，分别代表不同程度的agent自主权。agent模式完全由agent决定每一轮中何时添加和删除哪些工具，并且调用工具生成response；workflow模式则采用固定流程，由系统在每轮query之后先删除和添加工具，agent只负责调用工具生成response；Hybrid模式是混合模式，介于两者之间，作者发现agent移除工具的能力比添加工具的差，所以让系统负责移除工具，agent负责动态添加工具。

背景

本文的应用场景是agent在多轮对话(multi-turn conversation)中使用dynamic tool calling来回答用户query。dynamic tool calling就是function calling，指的是包含几百上千个函数/api/mcp server/工具，这点和我们之前读过的TIR类型工作只包含几个固定的(fixed)工具是不一样的，作者想解决的问题是面对这么多的工具，由于llm context size限制，如何根据每一轮的query，选择一个小的最合适的工具集合来augment llm生成response，为此提出了MemTool框架。

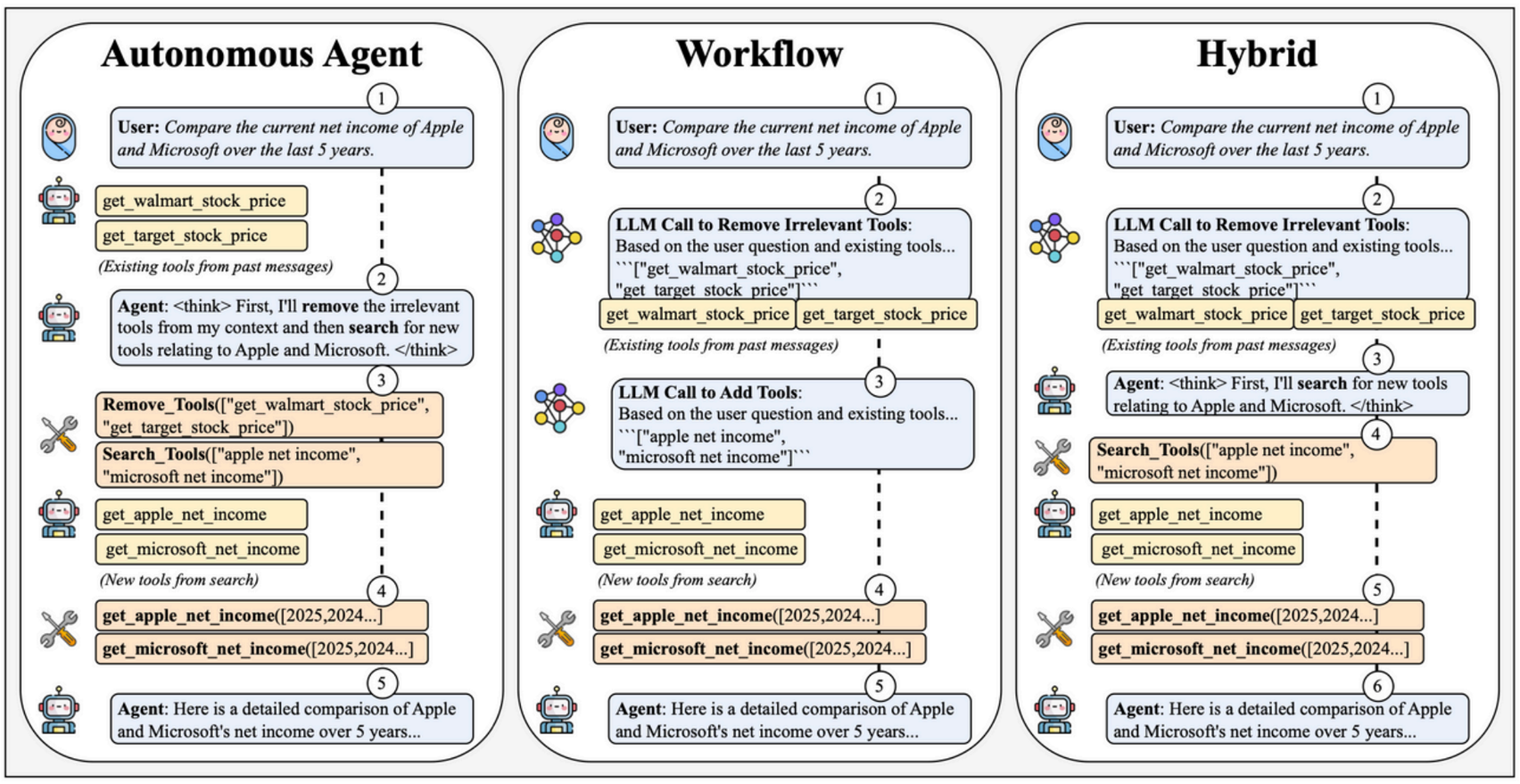
实验设置

- 实验数据集：ScaleMCP，包含5k个工具/MCP server，每一轮平均需要调用5次工具
- 多轮对话的轮数是100
- 评估了13个llm
- thinking model的工具管理能力比non-thinking强，特别是删除工具的能力，因此自主agent模式对lrm的要求比较高
- workflow模式很实用，hybrid模式则中和了两种mode，有一定的灵活性

MemTool的三种Mode

MemTool内置三种不同的工作模式，目的是找到最适合本轮query的工具集，对于每一轮query：

- 纯agent 模式：agent自主决定增加和删除工具
- workflow模式：让第三方llm增加和删除工具，agent不做这件事
- 混合模式：让第三方llm删除工具，agent可以自主决定增加哪些工具



思考

对我来说，读完本文最大的收获是终于找到一个合适的术语，来区分TIR中的tool calling和tool-augmented LLM时代的function calling了，就是dynamic vs fixed tool calling，一方面是使用的工具集合是否预先设置好，二是所涉及的工具数量也差别巨大，一般TIR只涉及几个工具，dynamic tool calling可以包含几百上千个工具/函数/MCP server。