**TPTU: Large Language Model-based AI Agents for Task Planning and Tool Usage**

[**https://arxiv.org/pdf/2308.03427**](https://arxiv.org/pdf/2308.03427)

**Problem**

1. LLM is incompetent in solving logical problems
2. training data is out of date
3. hallucinating
4. challenge remains in integrate tools and models into complex system

**Aim:** provide a helpful resource for researchers and practitioners to leverage the power of LLMs in their AI Task Planning and Tool Usage (TPTU) applications

**Contribution**

1. 提出了LLM-based AI代理框架，以评估LLM的TPTU能力
2. 设计Once-for-all 和 sequential manner 代理。提供了它们的实证结果和分析。
3. 提出了LLM的弱点：不能输出特定格式； 难以掌握任务需求；过度使用一个工具；缺乏总结能力，和见解。

**Methodology**

* AI Agent Framework：

1. Task Instruction: the instructions from human
2. Designed Prompt: includes system instructions, tool descriptions, few-shot demonstrations, chat history, or even error output.
3. Tool Set: external systems, like databases, APIs, tools, models. Agents interact with them to access or process information beyond its (LLM) internal knowledge, like current time, weather.
4. LLM: Core of the agent to interact with others, and generate intermediate output and final answer
5. Intermediate Output: (1) the **high-level plans** to fulfill the original user instruction, (2) selected and created **tools** to fulfill each subtask in the plans, and (3) the **results or errors** produced after tool execution.
6. Final Answer: summarized result

* Agent Ability: Percepting instruction; Task planning (generating sub-tasks, adjusting by changes and environment); Tool Usage (use or create tools); adjusting plans by Learning/Reflecting/Memory from feedback; summarization.
* one-step agent TPTU-OA: 将任务分解为一系列子任务。强调整体理解和规划
* Sequential agent TPTU-SA：成功解决子任务后请求LLM提供下一子任务。强调持续反馈。

**Evaluation**

* factors: the number of tools and the specific tools。 在一个场景下最多使用2个工具（但可多次采用）。
* 主要采用SQL生成器和Python生成器。此外还描述了10个其他工具（作为干扰）
* LLMs采用GPT-200B，Claude，InternLM-120B，Ziya-13B，ChatGLM-130B，Chinese-Alpaca-Plus-33B。

Evaluation on Task Planning:

* Evaluation on TPTU-OA:

1. metrics：规划工具使用顺序与ground truth使用顺序。结论：所有LLM都具有规划能力。
2. 修改prompts中的例子（增加对每个工具的使用说明）。metrics：两个序列都成功规划时（满足3点：1. tool顺序，2.第二个序列中的tool说明正确，3.说明的顺序与tool的顺序匹配）。 结论：所有LLM都下降了。见解：1.错误跟踪和调试困难；2.工具和子任务是独立生成的，难匹配；3.缺少灵活性；4.某些问题在未解决的早期不能提供有用信息。
3. 采用{Tool : sub-task}结构使规划Tool顺序时也生成相应的子任务。 metrics：与2相同。 结论：Claude提高了65%，其他也有提高。这种字典结构使工具与子任务绑定，可解释相对于2中的整体表现的提高。
4. 加入无关10个tools在prompts中。metrics：同2. 结论：评估结果显示保持不变/仅边际下降。这说明LLM-based AI agent能避开这些不想管工具，也证明了所设计的prompts的有效性。

* Evaluation on TPTU-SA: 继续使用tool-subtask pair的字典结构（因为已经验证了有效）。设计SA的prompts (加入history，Tool-subtask一次仅一组). metrics：同2.  结论：与3相比SA普遍优于OA，特别是高性能LLM（GPT，claude，和intareLM）。 见解：1. 现实中人类更倾向一步一步解决复杂问题。2.更丰富的上下文理解（更多信息），因为执行下一个任务时，上一个任务的结果被观察到。3.灵活性，因为后续步骤有机会调整策略或纠错。4.从历史中学习因为包含整个行动和结果的历史。

Evaluation on Tool usage:

* 首先评估SQL的生成。Appendix A中分别是人工设计的SQL数据库中表的列名和类别，以及设计的问题，对应的答案和SQL code例子。SQL数据集共有三类：1. Simple SQL queries (only basic operations)， 2.Complex nested SQL queries (nested queries, across multiple tables) 3. SQL和python结合的问题（旨在探索agent规划工具的能力）。
* 根据数据集设计新的prompts（利用CoT, 给出例子，要求模型输出Thought）。metrics：根据查询的响应评估。 结论：每个模型都展示出了使用工具的能力，该结果作为baseline。
* 根据数据集中complex nest SQL修改prompts。并测试direct-based和CoT-based的agents能力。 metrics：同上。 结论：大型LLM（GPT，Claude）均表现好且两种方法没有差别。ChatGLM 表现为CoT略好，而Ziya和IntirmLM则标为CoT略差。这可能表明LLM在处理子任务之间的联系或顺序问题的能力不同。能力好的模型可以利用CoT从逐步推理中获益，而反之逐步推理将增加模型负担。因此LLM-based agent也需要根据LLM的特点进行定制。

except SQL, now evaluate Mathematical Code Creation：

* 设计新的prompts以使模型生成python代码解决数学问题。 mitrics: (没提，应该根据执行python的结果来评估)。 结论：GPT，Claude，InternLM 大模型体现出强大的数学能力。ZIya也表现成功。而ChatGLM显示出很大困难在处理数学方面。这也再次强调了LLM具有不同特点。
* 根据数据集中SQL与python的混合工具使用的数据集。设计新的prompts并采用TPTU-OA和TPTU-SA测试。metrics：根据调用各个工具生成的结果进行评估。 结论：在TPTU-OA中，GPT 50%远高于其他（IntrmLM 15%, 其余为0）。在TPTU-SA中，GPT和IntrmLM均略微提高到55%，和20%。其他仍为0. 这显示出LLM在使用外部工具处理复杂任务的能力上有显著差异。此外也显示SA的顺序解决问题比OA更有效。

可借鉴：

1. prompts的设计包含多种例子。包括一次性整体规划OA和逐步规划SA。这为以后设计prompts提供参考。
2. 对于与数据对齐的问题可以利用像python代码中字典的格式{Key : Value}。
3. 该实验展现出不同LLM在不同任务上的能力差异较大。这个调查结果可以为日后选择LLM提供见解。
4. Acknowledge部分提到使用了DNN-based Tool Scheduling by LLMs。说明很可能LLM-based agent生成tool和sub-task后需要一个DNN将{Tool ：sub-task}自然语言文本映射到执行tool的actions上。因此在论文1中，作者没有介绍，可能也使用了这样的映射。（这也符合对话中的DNN-based Policy model）