


AWORLD: DYNAMIC MULTI-AGENT SYSTEM WITH STABLE MANEUVERING FOR ROBUST GAIA PROBLEM SOLVING

基于Execution-Guard的multi-agent架构

Zhitian Xie, Qintong Wu, Chengyue Yu, Chenyi Zhuang, Jinjie Gu

{xiezhitian.xzt, qintong.wqt, yuchengyue.ycy, chenyi.zcy, jinjie.gujj}@antgroup.com

代码地址: <https://github.com/inclusionAI/AWorld>

 AWorld Team, Inclusion AI

简介

本文提出包含两个agent的multi-agent系统: Execution Agent - Guard Agent。整体框架比较简单, 其中Execution Agent是主要的Agent, 负责将任务拆解为子任务, 调用各类外部工具来帮助推理完成任务; Guard Agent的作用是对Execution Agent的reasoning trajectory进行诊断找到问题甚至是可能存在的推理隐患, 帮助Execution Agent更好的推理。

那么Guard何时参与到Execution的reasoning trajectory中呢? 从prompt来看, Execution有固定的workflow, 其中包含一个名为Thinking Process Reviewing的阶段, 在这个阶段Execution会调用Guard来分析自己的trajectory。

背景

本文属于multi-agent方向的工作, 作者说受到船在航行过程中受到风浪影响, 从而需要依靠船舵不断修正航向的机制启发, 为multi-agent系统提出了动态机动(dynamic maneuvering)的机制, 让guard agent为主要的agent(execution agent)服务, 分析和纠正execution agent的reasoning trajectory中存在的问题或可能存在的推理隐患, 从而让execution agent更好的推理。

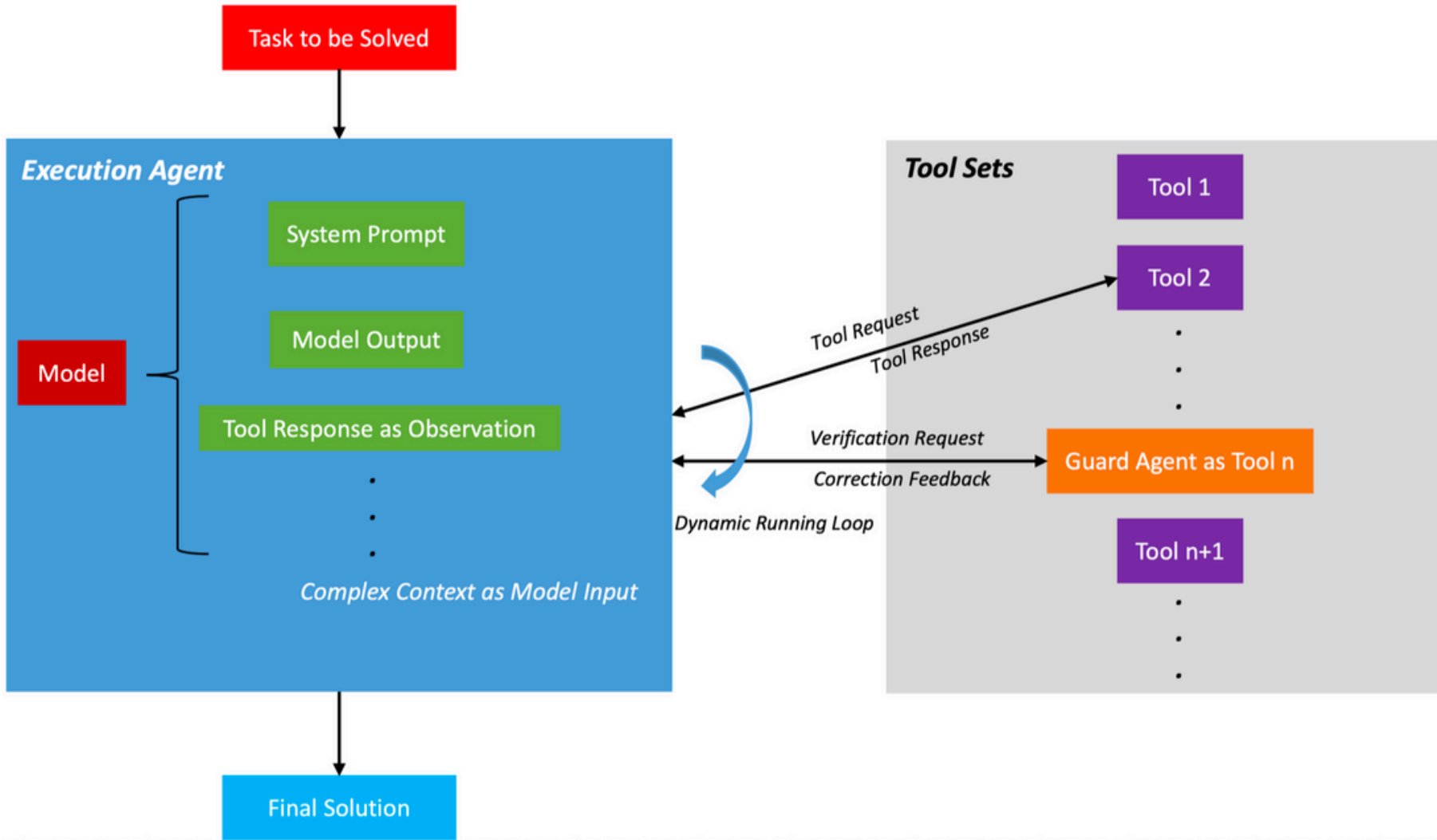
实验设置

- 评测集: GAIA
- 模型: Gemini 2.5 Pro

	Gemini 2.5 Pro	SAS	Gemini 2.5 Pro vs SAS	MAS	SAS vs MAS
Round 1 Pass@1	32.11%	57.8%		71.56%	
Round 2 Pass@1	30.28%	64.22%		65.14%	
Round 3 Pass@1	32.11%	65.14%		66.97%	
Pass@3	38.53%	81.65%	+111.91%	83.49%	+2.25%
Pass@1_avg	31.5%	62.39%	+98.06%	67.89%	+8.82%
Pass@1_std	0.00863	0.03265	+278.33%	0.02701	-17.3%

Table 2: Summary of experimental results across different versions.

Multi-Agent架构



思考

- 1.本文说是受到vessel maneuvering的启发, 但是在系统设计中并没有利用到相关的数学/技术, 所以我个人认为这更像是一种写作包装 (非贬义)
- 2.为什么不用自家的llm做实验呢🤔
- 3.在实验环节, baseline不太充分, 缺少带有动态调整机制的single-agent以及其他multi-agent方案作为对比, 当然不能否认本文的效果, 毕竟在GAIA leaderboard的排名是实打实的
- 4.关于调用guard的时机, 从prompt来看, 似乎并不是让execution在每一个step之后都调用guard, 而是仅在Thinking Process Reviewing阶段调用一次? 这部分没太看懂
- 5.execution-guard和更常见的critic-refine的区别, 我理解后者是得到完整的response(reasoning trajectory)之后, critic再参与批评, 前者是让guard在execution推理过程中就参与进来