arxiv.org/abs/2307.16789

TOOLLLM: FACILITATING LARGE LANGUAGE MODELS TO MASTER 16000+ REAL-WORLD APIS

开源代码：github.com/OpenBMB/ToolBench

Yujia Qin[1*], Shihao Liang[1*], Yining Ye[1], Kunlun Zhu[1], Lan Yan[1], Yaxi Lu[1], Yankai Lin[3†], Xin Cong[1], Xiangru Tang[4], Bill Qian[4], Sihan Zhao[1], Lauren Hong[1], Runchu Tian[1], Ruobing Xie[5], Jie Zhou[5], Mark Gerstein[4], Dahai Li[2,6], Zhiyuan Liu[1†], Maosong Sun[1†]
[1]Tsinghua University [2]ModelBest Inc. [3]Renmin University of China
[4]Yale University [5]WeChat AI, Tencent Inc. [6]Zhihu Inc.
yujiaqin16@gmail.com

简介

本文提出了ToolLLM，一套完整的工具使用(tool-use)框架，目的是提升开源LLM在真实任务中的tool calling 能力。ToolLLM涵盖tool-use质量指令微调数据创建、sft LLM以及评估全流程。核心贡献是构建了高质量、非常diversity、包括single-tool和multi-tool的数据集ToolBench，覆盖16000+个真实RESTful API。数据构建过程分为三阶段：先从RapidAPI收集并筛选真实API和文档，再利用ChatGPT生成指令(任务)和API组合，最后利用ChatGPT的function calling与DFSDT算法自动生成解决路径。
可以看到整个数据创建过程，主要依赖ChatGPT，减少了人工参与度。

背景

尽管开源LLM在指令微调(instruction tuning)方面也有显著的提升，但现有训练仍主要集中在语言相关的任务上，对LLM工具使用(tool-use)能力的训练明显不足。相比之下，闭源的SOTA模型在多轮调用搜索引擎、使用函数接口等实际任务中展现出强大的tool calling能力，但他们实现方式并不公开。
为弥补这一差距、加速开源LLM在tool-use 方向的发展，作者提出了ToolLLM——一个覆盖从tool-use指令微调数据构建、LLM tuning到评估的全流程tool-use框架，为开源模型掌握真实世界工具的调用提供了全面可复现的解决方案。

ToolLLM

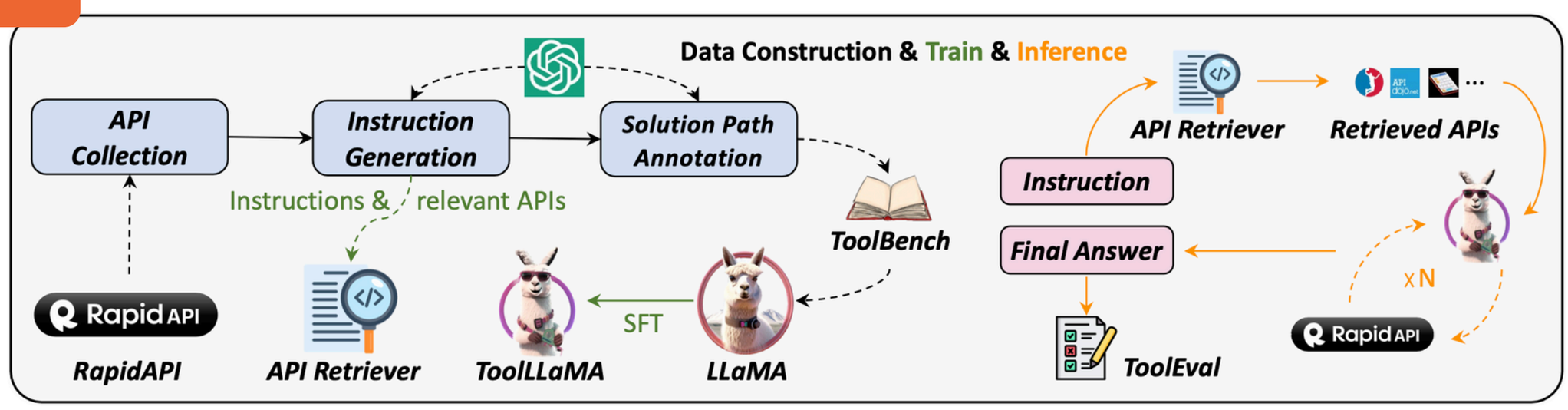

Figure 1: Three phases of constructing ToolBench and how we train our API retriever and ToolLLaMA. During inference of an instruction, the API retriever recommends relevant APIs to ToolLLaMA, which performs multiple rounds of API calls to derive the final answer. The whole reasoning process is evaluated by ToolEval.

部分实验结果

Model	Method	I1-Inst.		I1-Tool		I1-Cat.		I2-Inst.		I2-Cat.		I3-Inst.		Average	
		Pass	Win	Pass	Win	Pass	Win	Pass	Win	Pass	Win	Pass	Win	Pass	Win
ChatGPT	ReACT	41.5	-	44.0	-	44.5	-	42.5	-	46.5	-	22.0	-	40.2	-
	DFSDT	54.5	60.5	65.0	62.0	60.5	57.3	75.0	72.0	71.5	**64.8**	62.0	69.0	64.8	64.3
Claude-2	ReACT	5.5	31.0	3.5	27.8	5.5	33.8	6.0	35.0	6.0	31.5	14.0	47.5	6.8	34.4
	DFSDT	20.5	38.0	31.0	44.3	18.5	43.3	17.0	36.8	20.5	33.5	28.0	65.0	22.6	43.5
Text-Davinci-003	ReACT	12.0	28.5	20.0	35.3	20.0	31.0	8.5	29.8	14.5	29.8	24.0	45.0	16.5	33.2
	DFSDT	43.5	40.3	44.0	43.8	46.0	46.8	37.0	40.5	42.0	43.3	46.0	63.0	43.1	46.3
GPT4	ReACT	53.5	60.0	50.0	58.8	53.5	63.5	67.0	65.8	72.0	60.3	47.0	78.0	57.2	64.4
	DFSDT	60.0	**67.5**	**71.5**	**67.8**	**67.0**	**66.5**	79.5	**73.3**	**77.5**	63.3	**71.0**	**84.0**	**71.1**	**70.4**
Vicuna	ReACT & DFSDT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Alpaca	ReACT & DFSDT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ToolLLaMA	ReACT	25.0	45.0	29.0	42.0	33.0	47.5	30.5	50.8	31.5	41.8	25.0	55.0	29.0	47.0
	DFSDT	57.0	55.0	61.0	55.3	62.0	54.5	77.0	68.5	77.0	58.0	66.0	69.0	66.7	60.0
	DFSDT-Retriever	**64.0**	62.3	64.0	59.0	60.5	55.0	**81.5**	68.5	68.5	60.8	65.0	73.0	67.3	63.1

指令生成

构造ToolBench的第二阶段，先先从所有API中采样一组API组合，然后让ChatGPT 为这些 API 写出合理的自然语言任务指令。有点反着来的感觉，拿着钉子找锤子，挺有意思的。
至于DFSDT，说实话第一遍没看懂，先略过吧。

思考

本文的api(tool)数量不再是一个、两个或几个，竟然高达16k+，当然这也为创建训练集带来了极大的难度，依靠人力是不可能了，使劲薅ChatGPT羊毛，如何创建大规模tool的训练集，确实是一个难题。