Tool-Star: Empowering LLM-Brained Multi-Tool Reasoner via Reinforcement Learning

在RLVR背景下，如何让LLM学会同时使用多种TOOL帮助推理?

开源代码：HTTPS://GITHUB.COM/DONGGUANTING/TOOL-STAR

Guanting Dong[1], **Yifei Chen**[1], **Xiaoxi Li**[1], **Jiajie Jin**[1], **Hongjin Qian**[2], **Yutao Zhu**[1]
Hangyu Mao[3], **Guorui Zhou**[3], **Zhicheng Dou**[1]*, **Ji-Rong Wen**[1]
[1]Renmin University of China [2]BAAI [3]Kuaishou Technology
{dongguanting, dou}@ruc.edu.cn

简介

本文提出了Tool-Star，将Tool-Integrated Reasoning (TIR) 扩展到Multi-Tool，要解决的第一个问题是如何构造多工具推理训练集（多阶段数据合成）？第二个问题面对如此复杂的多工具推理场景，如何设计reward function（层级reward）？以及如何用RL训练（SFT+self-critic RL）？

背景

什么是多工具(Multi-Tool)推理？指的是 LLM 在进行推理时，不仅使用一个外部工具（如搜索引擎、计算器），而是可以根据需求，在多个工具之间切换与协同使用。
举个例子，一个问题需要先搜索，再思考，再分析搜索返回的网页内容，再思考，再调用搜索引擎，再调用计算器，最后思考得到答案。
可见，相比单工具场景，问题要复杂的多。

实验设置

- 如何构造训练集？说实话过程比较复杂，共包含3个阶段的数据处理，简单来说，作者通过Prompt和hint-based sampling两类方法生成以及从没有使用tool的推理数据扩展成结合tool的推理数据。然后进行过滤，再按照课程学习思想，对推理数据的难易程度排序，后面训练模型时，简单数据用于sft，复杂数据用于RL。
- reward function：包括format reward、答案是否正确、如果format和答案都正确看推理时是否用了多种工具（这是为了鼓励模型调用多种tool）

$$R = \begin{cases} \max(Acc. + r_{\mathrm{M}}, Acc.) & \text{If Format is Good \& Acc.>0} \\ 0 & \text{If Format is Good \& Acc.=0} \\ -1 & \text{Otherwise} \end{cases}, r_{\mathrm{M}} = \begin{cases} 0.1 & \text{If } \exists(\texttt{<search>}\,\&\,\texttt{<python>}) \\ 0 & \text{Otherwise} \end{cases}$$

多工具推理训练集的数据合成

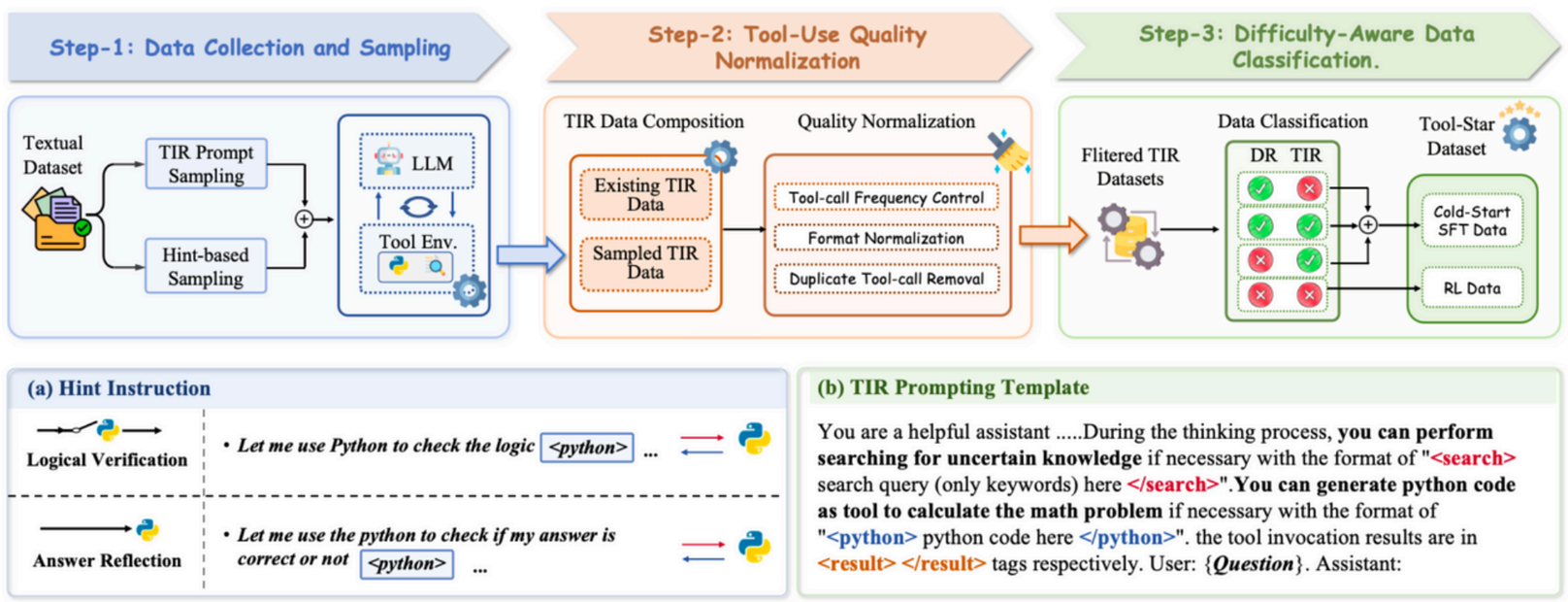

Figure 2: The overview of 3-step tool-integrated reasoning data synthesis pipeline.

多工具SELF-CRITIC RL训练流程

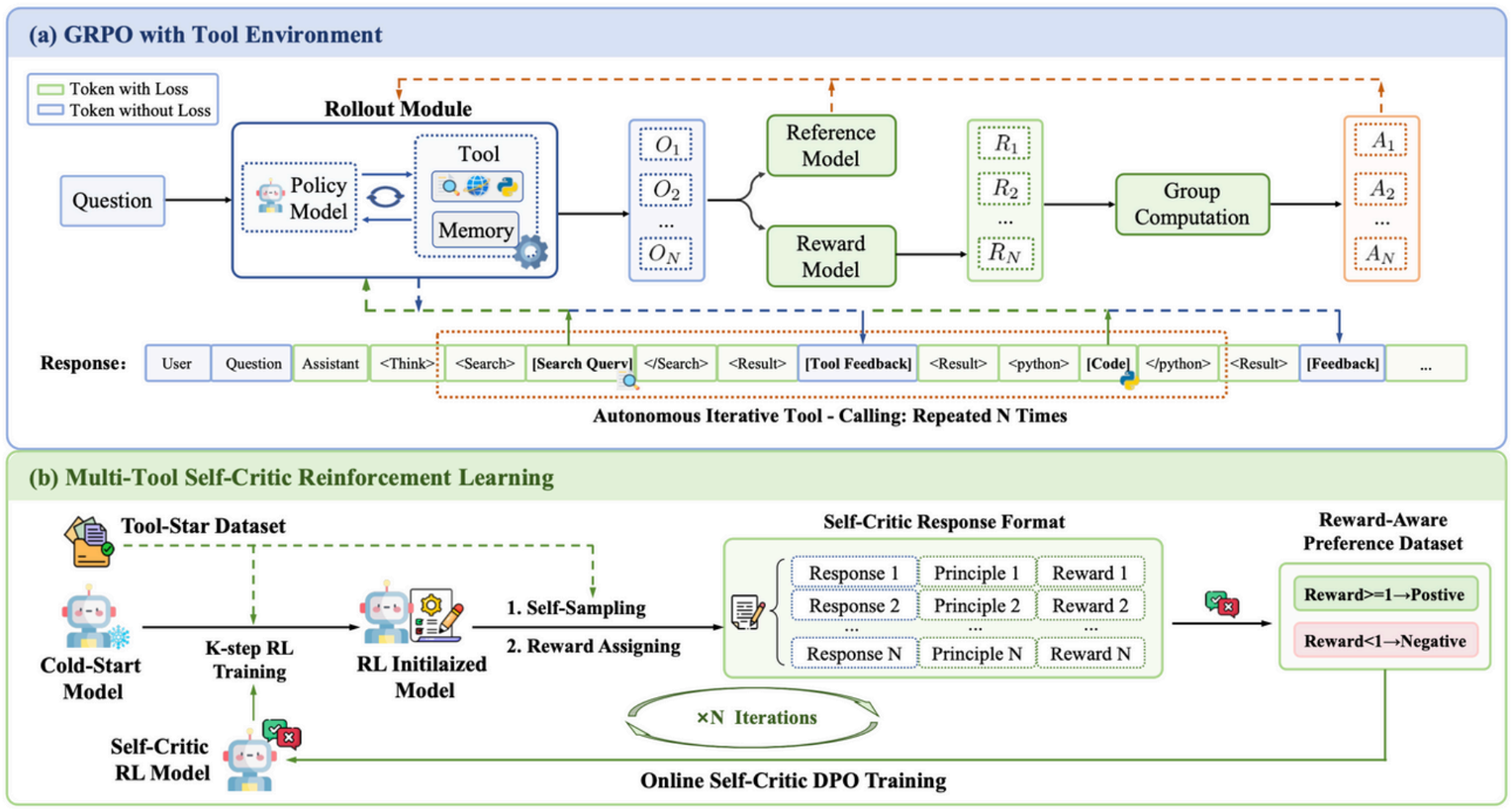

Figure 3: The overall framework of Multi-Tool Self-Critic Reinforcement Learning.

训练流程

作者设计了两阶段训练，1）SFT；
2）RL。考虑到REWARD的复杂性，RL阶段不只是使用GRPO来训练，还额外结合了SELF-CRITIC DPO。
为了提升多工具调用效率，作者还结合了缓存机制。

Algorithm 1 Multi-Tool Self-Critic Reinforcement Learning

Require: Reasoning Model π_θ, External Tools T, Reward Function $\mathcal{R}_\phi$, Cycle Times $\mathcal{C}$, GRPO Steps $\mathcal{S}$
Input: Datasets D, Task instruction I
Initialize $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
for $i \leftarrow \{1, ..., \mathcal{C}\}$ **do**
  for step $\leftarrow \{1, ..., \mathcal{S}\}$ **do** ▷ Start Vanilla RL phrase (GRPO)
    Sample a batch D_b from D
    for each query $q \in D_b$ **do**
      $q \leftarrow I \oplus q$
      Sample G sequences for q with T: $\{o_j\}_{j=1}^{G} \sim \pi_\theta(\cdot \mid q, T)$
      Calculate the optimization objective function J_θ of GRPO according to Eq.(4)
      Update the weight parameters of π_θ: $\theta \leftarrow \theta + \eta \cdot \nabla_\theta J_\theta$
  Randomly sample D_{sample} from D ▷ Start Self-Critic DPO
  for each query $q \in D_{sample}$ **do**
    $q \leftarrow I \oplus q$
    Sample the query q for G times with T to generate G different results: $\{o_j\}_{j=1}^{G} \sim \pi_\theta(\cdot \mid q, T)$
    Filter one correct result and one incorrect result from G results to obtain o_{chosen} and o_{reject}
    Calculate the optimization objective function J_θ of DPO according to Eq.(5)
    Update the weight parameters of π_θ: $\theta \leftarrow \theta + \eta \cdot \nabla_\theta J_\theta$
Output: Trained Model $\pi_{\theta_{\text{new}}}$

思考

从单独一个工具调用到多工具推理，是LLM TOOL CALLING的必然发展路线，本文做了一次很好的尝试，限于篇幅，这里就不多啰嗦了，后续可能会写一篇长文总结多工具推理。

@机器爱学习