Training Language Model to Critique for Better Refinement

arxiv.org/abs/2506.22157

Tianshu Yu[1,2*], Chao Xiang[1*], Mingchuan Yang[1], Pei Ke[3], Bosi Wen[2†], Cunxiang Wang[4,5], Jiale Cheng[2†], Li Zhang[1], Xinyu Mu[1], Chuxiong Sun[1], Minlie Huang[2‡]

[1]China Telecom Research Institute

[2]The Conversational Artificial Intelligence (CoAI) Group, Tsinghua University

[3]University of Electronic Science and Technology of China

[4]The Knowledge Engineering Group (KEG), Tsinghua University

[5]Zhipu AI

dailyyulun@gmail.com aihuang@tsinghua.edu.cn

开源代码：https://github.com/publicstaticvo/critique

RCO: 面向改进(refine)的批评模型(critic model)优化

背景

当前关于critic model的研究主要有两个方向：

1)）通过微调llm得到critic model，用于协助人类进行evaluation，比如OpenAI的两个典型工作，分别用sft和rlhf tuning 得到critic model，期望用来解决scalable oversight问题；

2）在许多llm workflow和agent系统中，普遍包含critic-refine流程，即先由critic model对llm输出进行批评（critique），再根据批评进行改进（refine）以提升回答质量。然而，这类流程通常依赖prompt engineering，将现有的llm直接充当critic使用，并未针对critic model 进行专门微调优化。

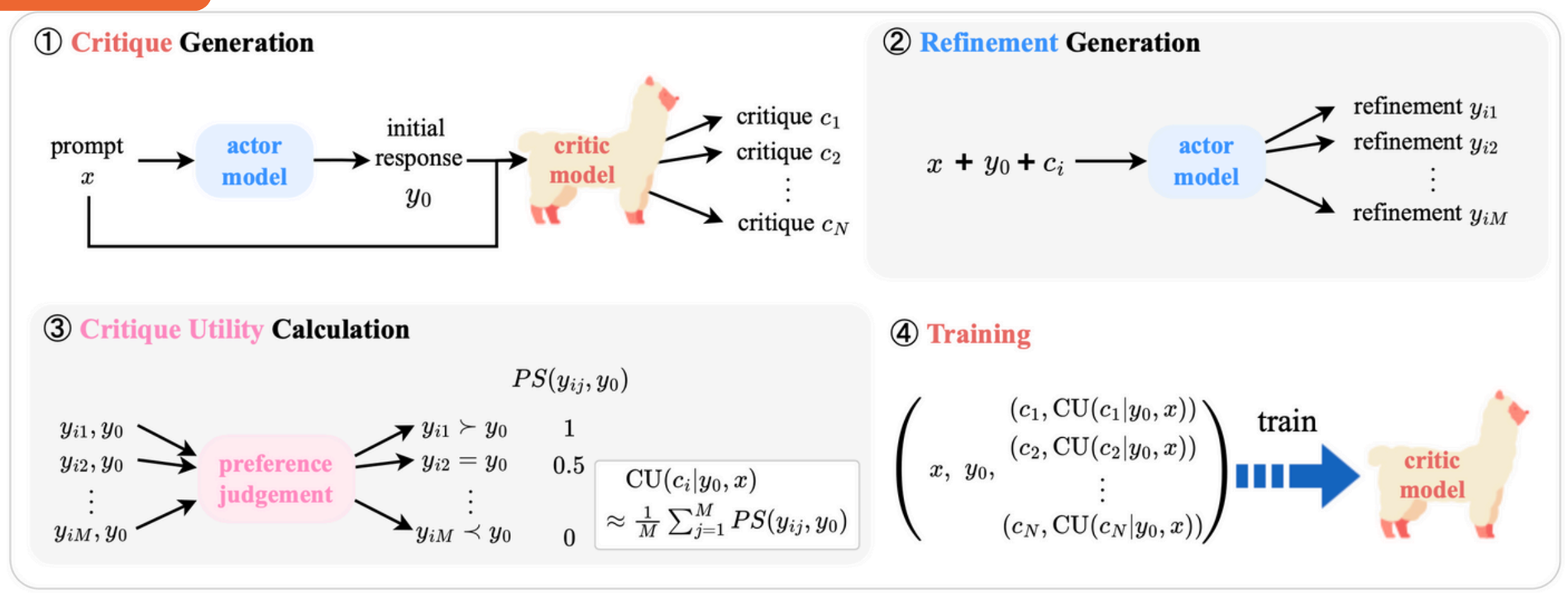
本文以第二种场景为切入点，提出在critic-refine 流程中，通过训练得到更好的critic Model，使其生成的批评更具可操作性和针对性，从而更有效地提升refinement阶段的回答质量，最终优化整体系统效果。

简介

本文提出Refinement-oriented Critique Optimization (RCO)，它是以优化(refinement)为导向的训练critic model的强化学习算法，目的是在critic-refine流程中有效训练critic model，让critic model生成有助于回答改进的批评(critique)，从而提升refine效果。

简单来说，RCO是以DeepMind提出的强化学习算法Direct Reward Optimisation (DRO)为基础，同时采用RLVR的做法，作者设计了critique utility (CU, 批评效用) 作为可验证的reward值，即通过judge model比较refinement后回答与原回答的优劣计算得到CU值。

RCO训练流程



整体框架包含三个llm: actor model, critic model和judge model.

根据prompt x, actor model先生成初始的response y_0, 然后把(x, y_0)输入给critic model生成N个不同的critique;

对于每一个critique c_i, 把(x, y_0, c_i)输入给actor生成M个新的response y_{ij}, 也就是refinement结果;

judge model比较y_0和y_{ij}哪个更好，对于每个critique c_i, 计算批评效用(CU)值：$CU(c_i|y_0,x) = \frac{1}{M}\sum_{j=1}^{M} PS(y_{ij}, y_0)$

基于RL算法Direct Reward Optimisation (DRO)来训练critic model. 对应论文公式(3)-(7)，如果我没有理解错的话，这部分就是DRO算法，只是用CU作为reward 值。

思考

首先，在critic-refinement流程中通过RL tuning得到高质量的critic model，肯定是很有意义的，关键是如何设计reward function？毕竟critic model的输出critique不是直接作为整体流程的输出，而是中间产物，需要提供给llm做refinement, 而refinement结果才是流程的输出，因此难以直接评估critique质量来设计reward function，所以本文设计reward的方式还挺值得学习的。

另外，我联想到在机器学习时代，一个有用的trick是把系统评估指标作为优化目标函数，似乎和本文面向refinement的critic model训练优化也能对应起来。

@机器爱学习