arxiv.org/abs/2506.18841

LongWriter-Zero: Mastering Ultra-Long Text Generation via Reinforcement Learning

开源代码：github.com/THUDM/LongWriter

LongWriter-Zero: 用RLVR的方式训练reasoning llm进行长文生成任务

Yuhao Wu[1]*, **Yushi Bai**[2]*, **Zhiqiang Hu**[1], **Roy Ka-Wei Lee**[1], **Juanzi Li**[2]

[1]Singapore University of Technology and Design, Singapore
[2]Tsinghua University, Beijing, China

简介

本文关注如何让reasoning llm用于长文生成(long-form generation)任务，围绕以下三个问题展开研究：1) 如何训练能做长文生成的reasoning llm? 作者使用RLVR思路，核心是设计包含Length RM、Writing RM、Format RM 三项的reward function，实现从零激活base llm的长文生成能力; 2) 在长文生成任务上，reasoning llm是否也呈现 test-time scaling？作者实验验证用包含显式Think阶段的prompt比直接answer的prompt效果更好; 3) 最近的一些工作表明reasoning llm的能力天花板其实受到base model制约，在长文生成任务上是否也如此呢？作者发现对base llm继续预训练(continual pretraining)再RL训练，可提高RL训练的起点和最终表现，说明RL能激活但无法突破base model能力限制。

背景

目前llm的长文生成能力不足，我们前面读过的LongWriter提供了一个以sft为核心的方案，但sft依赖~~现成的(似乎还没看到过)或者~~合成的长response数据。受到reasoning llm启发，作者提出LongWriter-Zero，不用sft直接RL从零开始激活base llm的长文生成能力。

实验设置

- RL算法：GRPO
- 模型：Qwen2.5-32B base
- 训练集query采样自WildChat-1M和LMSYS-Chat-1M，作者设计了prompt template让QwQ-32B估算query的response长度，以此为依据筛选query

Reward设计

RLVR reward function:

- length reward: 评估response长度是否合适，ground truth是用qwq-32b预测的response范围

$$r_{\text{length}}(o) = \begin{cases} 1, & \text{if } L_{\text{lower}} \le len(o) \le L_{\text{upper}}, \\ \dfrac{len(o)}{L_{\text{lower}}}, & \text{if } len(o) < L_{\text{lower}}, \\ \dfrac{L_{\text{max}} - len(o)}{L_{\text{max}} - L_{\text{upper}}}, & \text{if } len(o) > L_{\text{upper}}. \end{cases}$$

- writing reward: 评估response的质量，用human preference 数据 tuning Qwen2.5-72B得到reward model
- format reward: 一方面是response结构完整，要符合<think> ... </think><answer> ... </answer>，另一方面是对重复生成进行惩罚
- 为了避免不同reward scale的影响，最终是对三个advantage计算均值

$$A_{\text{final}} = \frac{1}{3}\left(A_{\text{length}} + A_{\text{write}} + A_{\text{format}}\right),$$

思考

长文生成任务用reasoning llm是否更有优势？对于这个问题我也很好奇，像数学/编程领域，llm先思考再解答直觉上是合理的，一是因为人类就是这样做的，二是直接用RLVR就能激活llm的思考能力，不需要人的先验参与。那么长文生成呢？所以作者提出的三个研究问题还是很合理的，如何训reasoning llm? 是否存在test-time scaling? reasoning llm能力是否受到base 决定?
当然在长文生成任务中设计reward function肯定要比数学/编程复杂多了，因为难以量化，比如多大的response length是ground truth呢？本文用qwq-32b的预测范围作为监督信号，估计噪声不小，也是无奈之举。另外writing reward部分用的是一个reward model而不是通过function计算的reward值，我暂且将本文的RL称为RLVR吧。再就是验证test-time scaling用的是prompt中是否要llm先think再answer，也有点难以信服（至少对我来说），我更想看到效果增长趋势而不是用有和没有<think>的差异来验证scaling law。

@机器爱学习