R^{2}EC: Towards Large Recommender Models with Reasoning

如何基于LLM构建统一的REASONING-THEN-RECOMMEND推荐模型?

开源代码：HTTPS://GITHUB.COM/YRYANGANG/RREC

Runyang You[1] Yongqi Li[1]* Xinyu Lin[2] Xin Zhang[1]
Wenjie Wang[3] Wenjie Li[1] Liqiang Nie[4]

[1]The Hong Kong Polytechnic University [2]National University of Singapore
[3]University of Science and Technology of China [4]Harbin Institute of Technology (Shenzhen)

runyang.y@outlook.com, liyongqi0@gmail.com,
xylin1028@gmail.com, zhangxin2023@stu.hit.edu.cn,
wenjiewang96@gmail.com, cswjli@comp.polyu.edu.hk, nieliqiang@gmail.com

简介

本文提出R2ec，一个基于LLM的统一推荐模型，模型先生成语言推理过程，再进行目标item推荐，通过将推理生成与推荐决策融合为一条推理轨迹，实现了Reasoning-then-Recommend。R2ec构建于一个共享backbone(LLM)的多任务框架之上，推理与推荐共享参数并协同训练，为实现有效训练，作者设计了多路径采样机制，所有路径参与推理模块更新，而只用优势最大的推理路径进行推荐head优化。

背景

如何借助LLM的推理能力提升推荐效果？目前主流做法通常是将语言推理与推荐任务分开处理，分别建模，难以进行统一优化，模型是否能真正学会"推理驱动推荐"，要打个问号。为此，本文提出R2EC，将LLM推理和推荐两个差异很大的任务融合到同一个模型中，实现了"先推理，再推荐"(REASONING-THEN-RECOMMEND）的端到端训练。

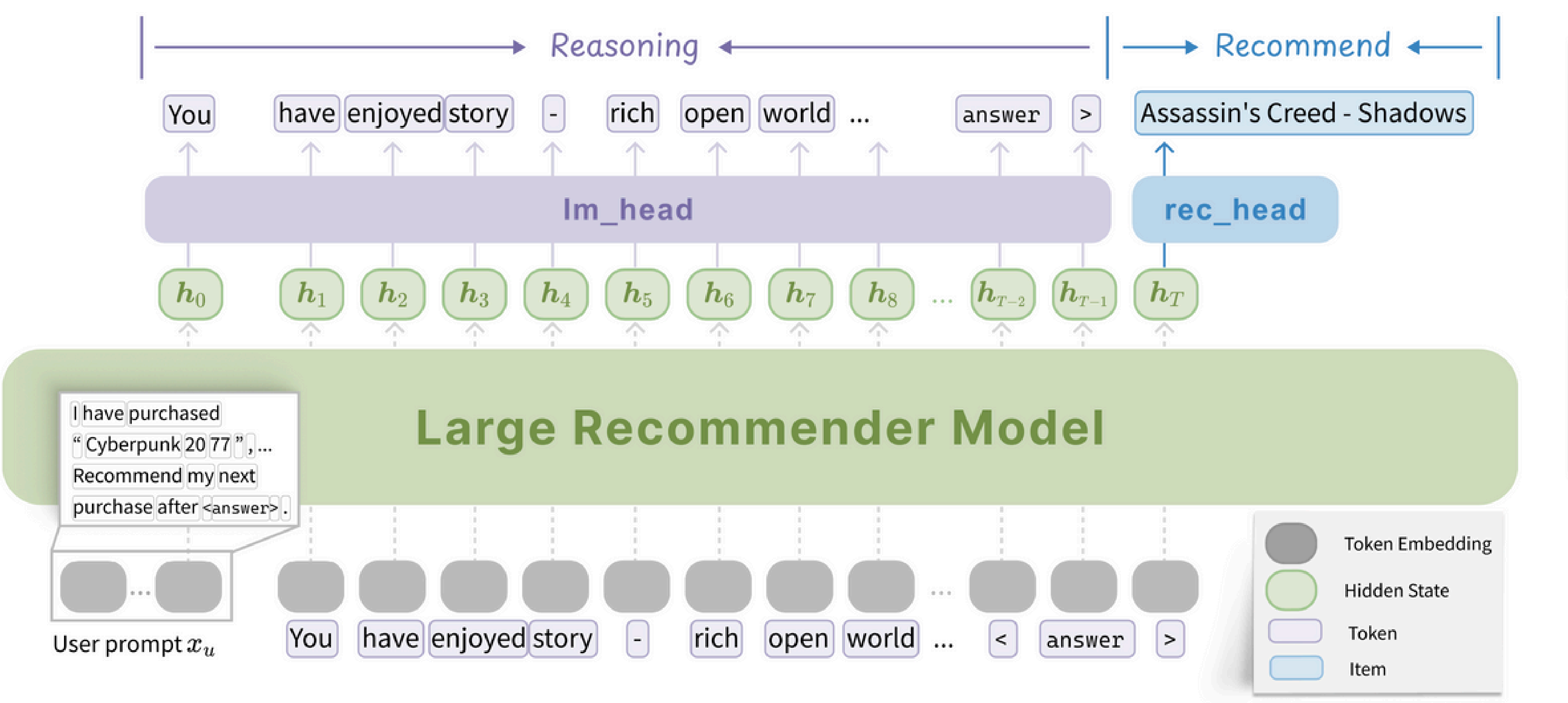
实验设置

- RL框架：trl，RL算法：RecPO，LLM: Qwen2.5-3B-Instruct和Gemma-2-2b-it
- Reward function：包含NDCG和in-batch softmax两项。后者权重很小(0.05)

$$R_d = \mathrm{NDCG}\,@k\,(\mathrm{rank}(v^+)) \qquad R_c = \frac{\exp(\boldsymbol{h}_T^\top \boldsymbol{h}_{v^+}/\tau)}{\sum_{v\in\mathcal{V}} \exp(\boldsymbol{h}_T^\top \boldsymbol{h}_v/\tau)},$$

$$R = \beta\, R_c + (1-\beta)\, R_d, \qquad \beta \in [0,1],$$

模型结构



USER和ITEM PROMPT

User Prompt

Analyze in depth and finally recommend next {category} I might purchase inside <answer> and </answer> . For example, <answer> a product </answer> .

Below is my historical {category} purchases and ratings (out of 5):

```
{% for hist in purchase_histories %}
  {% {hist.time_delta} ago: [{hist.item_title}] ({hist.rating}) %}
```

Item Prompt

Summarize key attributes of the following {category} inside <answer> and </answer> :

```
{% for key, attr in item.meta %}
  {% {key}: {attr} %}
```

训练过程

Algorithm 1 Training Process

Input: Dataset $\mathcal{D}$, initial policy π_θ, embedding function f_θ, item embedding table $\mathbf{H}_\mathcal{V}$
Output: Optimized policy model π_θ

for step $= 1$ to N **do**
  if step % $T_{\text{refresh}} == 0$ **then**
    Refresh item embedding: $\mathbf{H}_\mathcal{V}[v] \leftarrow f_\theta(x_v), \quad \forall v \in \mathcal{V}$
  end if
  Sample a training batch $\mathcal{B} = \{(u, v^+)\} \sim \mathcal{D}$
  Encode target item prompts and update embedding table: $\mathbf{H}_\mathcal{V}[v^+] \leftarrow f_\theta(x_{v^+}) \quad \forall (u, v^+) \in \mathcal{B}$
  for all (u, v^+) in $\mathcal{B}$ **do**
    Generate G trajectory: $\{[o_1, v^+] ..., [o_G, v^+]\} \sim \pi_{\theta_{\text{old}}}(\cdot|x_u)$
    Compute reward for each trajectory using Eq. (5)
    Compute advantage for each trajectory using Eq. (2)
  end for
  Update policy parameters θ via loss in Eq. (8)
  Update old policy: $\theta_{\text{old}} \leftarrow \theta$
end for

思考

很久不做推荐了，不太确定自己的理解是否准确。RREC的核心目标是训练一个LLM，能够先生成推理过程再进行推荐。简单来说，就是将推理轨迹中最后一个TOKEN的隐状态输入到推荐HEAD，用于预测目标 ITEM。但LLM推理和推荐系统可是两个差异很大的任务，如何统一到同一个模型中进行训练呢？
作者设计了RECPO强化学习算法：首先采样多条推理路径，每条路径最后都推荐一个ITEM，然后分别计算融合奖励（推荐排序的 NDCG 分数 + 推理表示与目标 ITEM 的相似度），并据此估算每条路径的优势（ADVANTAGE）。所有路径都参与推理部分的训练，而推荐部分则仅使用优势最大的那条路径进行更新。