

## 2024阿里预测：小语言模型能成为序列推荐的强力Buff吗？



SmartMindAI

专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注

已关注

9 人赞同了该文章

### Introduction

序列推荐因其捕捉用户兴趣随时间变化的能力在互联网广泛应用，但受限于封闭数据集训练，常产生曝光和流行度偏差。因此，我们强调结合开放[世界知识](#)<sup>+</sup>，不仅利用有限的用户-商品交互数据，还强调对用户行为的全面和泛化理解。这样有助于超越现有模型，深化对历史行为的理解。基于LLMs的[推荐系统](#)<sup>+</sup>主要分为两型：

1. 直接应用：利用LLMs强大的推理，如GPT 3.5/4，作为排序模块，通过在大规模[语料库](#)<sup>+</sup>上训练，它们能够进行类人水平的思考和推理，克服封闭数据限制，提供更全面的推荐决策。
2. 知识蒸馏框架（SLIM）：通过将大模型如ChatGPT-175B的解释作为标签，对小型学生模型（如LLaMA2-7B）进行知识迁移，这是一种资源效率高、推理逐步的方法，以较低成本提供可接受的推荐理由。

如何在有限资源下充分利用LLMs的chain-of-thought（CoT）推理，既要保证推荐质量，又需避免误导。一方面，需发展策略以指导LLM逐步解析用户行为，使其精准理解并生成相关推荐。另一方面，面对LLM的大规模需求，应聚焦于成本效益高的解决方案，如通过知识蒸馏将大模型的知识传递给轻量级学生模型，如LLaMA2-7B，同时减少对GPU服务器和API调用的依赖。这要求我们在利用LLMs的智慧的同时，巧妙管理成本，以适应实际推荐系统的资源约束。

- 我们开创性地提出了一个专门针对LLMs的序列推荐知识蒸馏框架，该框架旨在优化资源利用，既能保持推荐性能，又能有效地利用LLMs的chain-of-thought（CoT）推理能力。这是首次在设计时就针对LLMs的特性来解决推荐系统在面对封闭数据和资源限制时的问题。通过这种方法，我们不仅解决了LLMs推理的准确性问题，还通过逐步知识转移降低了对昂贵资源的依赖，同时确保生成的推荐理由具有实际价值。
- 我们创新地推出了SLIM，这是一种独特的序列推荐系统，专注于通过知识蒸馏技术巧妙地整合[大型语言模型](#)<sup>+</sup>（如LLMs）的CoT推理。SLIM设计的目标是将LLMs的强大能力以一种资源节约的方式传递给推荐系统，同时确保推荐质量和避免误导。通过在大模型（如ChatGPT-175B）上使用CoT提示，结合对学生的轻量级模型（如LLaMA2-7B）进行解释导向的训练，我们实现了在有限资源下有效利用LLMs的推理优势。这种框架不仅提升了推荐的精确性，还通过成本效益高的方法生成了实用的推荐理由。

### The Proposed Framework

我们创新地提出了SLIM（Step-by-step Knowledge Distillation for Recommendation），一种专门针对推荐系统的新型知识蒸馏方法。它巧妙地利用大型语言模型（如GPT 3.5/4）的CoT（chain-of-thought）推理能力，通过在大模型（如ChatGPT-175B）上进行CoT提示的指导下，对小型学生模型（如LLaMA2-7B）进行逐步知识转移。

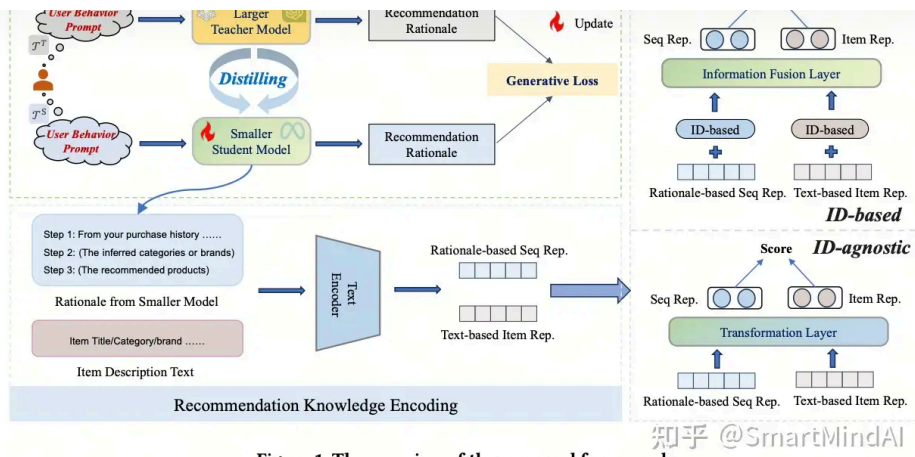


Figure 1: The overview of the proposed framework.

这种设计避免了直接应用大型模型带来的高昂资源消耗，同时保持了推荐的精确性和有效性。在实验中，我们证实了SLIM在多个数据集上表现优异，既能提供有意义的推理，又能以合理成本实现推荐服务。简而言之，SLIM是将LLMs的优势与推荐系统需求相结合的创新解决方案。

## Sequential Recommendation Backbone

$$\mathbf{z}_i = \text{ItemEncoder}(i, f_i)$$

在用户行为分析<sup>+</sup>中，我们构造用户 $\mathbf{u}$ 的行为序列

$$\mathcal{S}_u = [i_1, i_2, \dots, i_{t-1}]$$

其中每个交互项 $i_k$ 属于集合 $\mathcal{I}$ 。

我们通过

$$\mathbf{Z}_u = \text{ItemEncoder}(i_1, i_2, \dots, i_{t-1})$$

提取每个交互的物品表示 $\mathbf{Z}_u$ 汇集了所有交互的信息。接着，我们运用序列建模<sup>+</sup>技术，如LSTM或GRU，来对这些嵌入进行序列编码，目的是捕捉行为序列中的动态变化和关联性。LSTM和GRU能学习相邻项间的依赖关系，从而提升推荐的预测准确性。

$$\mathbf{s}_u = \text{SeqEncoder}(\mathbf{Z}_u)$$

在处理用户行为时，我们用 $\mathbf{s}_u$ 来表示用户序列 $\mathcal{S}_u$ 。然后，我们利用SeqEncoder对序列进行编码，可能包含注意力机制<sup>+</sup>或其他神经结构，以理解序列中的动态变化。目标是预测用户在步骤 $t$ 将选择的下一个交互项 $i_t$ 。

预测阶段，我们通过点积<sup>+</sup>或多层感知器<sup>+</sup>（MLP）加上sigmoid激活函数<sup>+</sup>，计算出预测向量 $\hat{\mathbf{y}} \in \{0, 1\}$ ，每个元素 $\hat{y}_{ui}$ 代表物品 $i$ 被推荐给用户 $\mathbf{u}$ 的概率。模型通过优化二元交叉熵<sup>+</sup>损失进行训练，以减小预测与实际标签的误差。

$$\mathcal{L} = - \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui})$$

传统的序列推荐模型主要依赖用户行为序列及物品特征（如标题、类别和品牌），如将用户过去的购买或浏览行为转化为预测下一个可能行动。然而，它们往往忽视了像LLMs这样的先进模型所具备的复杂推理能力，后者在理解和生成深层次关联方面显示出独特潜力。

## Step-by-Step Knowledge Distillation for Recommendation

尽管LLMs强大的推理能力显著提升，直接应用它们在推荐系统面临挑战：

- (1) 用户行为的复杂性超出了LLMs的理解范围；
- (2) 大模型的计算资源需求和潜在延迟问题。为克服这些问题，我们采用逐步知识蒸馏策略。

方式，我们以低成本获得LLMs的推理优势，为推荐系统提供基于推理的有价值建议。

Extracting Recommendation Rationales from LLMs

- 1. 兴趣探索：我们利用LLM解析用户过去的行动轨迹，去发掘他们可能未明确表达的兴趣领域。
- 2. 类别/品牌洞察：通过LLM的CoT思维，系统深入理解用户在行为中的类别和品牌偏爱。

精准推荐

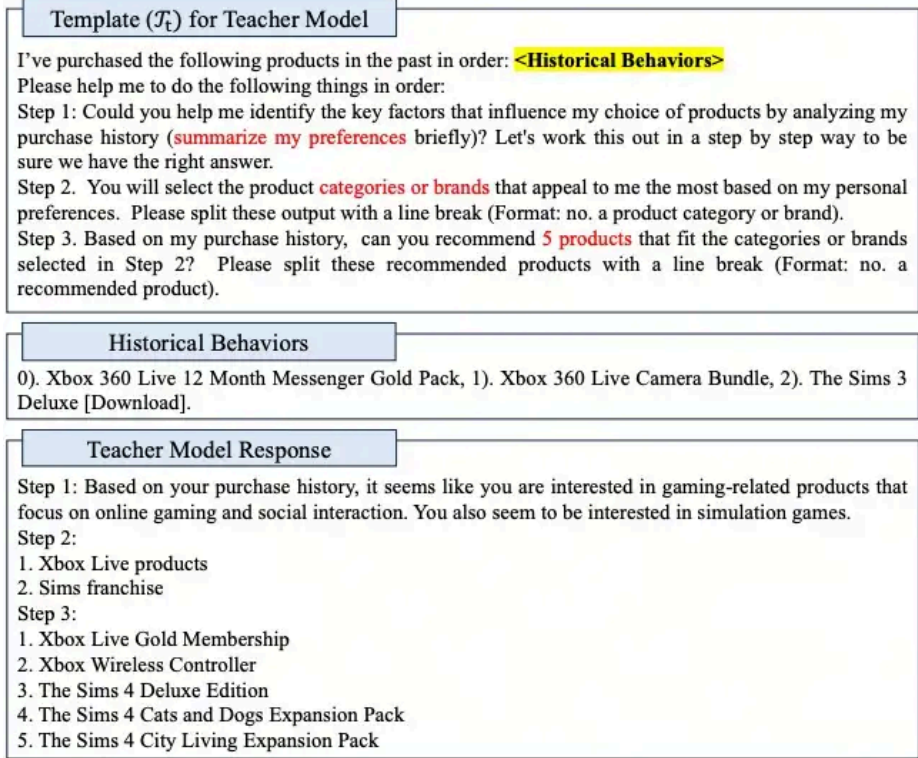


Figure 2: Zero-shot CoT prompting for larger teacher model. Eliciting LLMs to reason in a step-by-step manner.

- 1. 步骤1. 利用LLMs解析用户行为序列，提炼出用户的主要偏好模式。
- 2. 步骤2. 依据提炼出的用户偏好信息，针对性地向用户推荐相应的商品类别或品牌。
- 3. 步骤3. 结合前面的类别/品牌分析<sup>+</sup>，LLM为用户推荐与其已显示偏好相符合的具体商品，确保推荐内容贴合用户的真实需求。

利用用户历史行为的模式（如行为模板 $\mathcal{T}_t$ ），从小型数据子集 $\mathcal{D}'$ 中提取特定信息，构造教师LLM的CoT提示集 $\mathcal{X}'$ 。这些提示促使模型生成针对每个用户实例 $r'_u$ 的个性化推荐理由 $\mathcal{R}'$ 。尽管当前推荐系统未直接利用这些理由，但它们对提升透明度和理解力至关重要。由于多数实际商品（如消费品）与公开知识关联难度较大，我们避免依赖此类间接证据。

Fune-tuning Smaller Models with Recommendation Rationales

知乎

I've watched the following movies in the past in order: **<Historical Behaviors>**  
Please provide a **brief summary** of my products preferences, followed by the recommendation of five product categories or brands that align with my interests, and then suggest **five products** that fall under each category or brand.

#### Historical Behaviors

0). Gundam Breaker, 1). Shin Gundam Musou, 2). Forza Motorsport 4 - Xbox 360, 3). Armored Core: For Answer - Xbox 360, 4). Dynasty Warriors: Gundam 3 - Xbox 360, 5). Import Tuner Challenge - Xbox 360, 6). Forza Horizon - Xbox 360, 7). PlayStation Vita Memory Card 64GB (PCH-Z641J).

#### Student Model Response

Step 1: From your purchase history, it seems that you enjoy gaming-related products that are action-packed, have fast-paced gameplay, and involve warfare or fighting. You also seem to prefer games based on popular franchises such as Dynasty Warriors and Gundam. Additionally, you have a preference for racing and sports games.

Step 2:

1. Racing games
2. Action-packed games with fast-paced gameplay
3. Games based on popular franchises such as Dynasty Warriors and Gundam

Step 3:

1. Forza Horizon 5 - Xbox Series X/S
2. Gundam Breaker: Battlege - PlayStation Vita
3. Dynasty Warriors 9 - PlayStation 4
4. Need for Speed Heat - Xbox One
5. Gran Turismo Sport - PlayStation 4

知乎 @SmartMindAI

$$\min_{\theta} -\log P(r'_u | p'_u; \theta)$$

这里 $\theta$ 是模型参数 $r'_u$ 是针对用户实例的个性化推荐理由 $p'_u$ 是基于用户行为模板 $T_i$ 在小数据集 $D'$ 上生成的CoT提示，通过训练，我们寻求找到最优参数 $\theta$ ，使得生成的理由 $r'_u$ 最贴近用户真实需求，同时减少对计算和存储的需求，以满足动态推荐环境的要求。

$$\mathcal{L}_{distill} = \sum_{u \in \mathcal{U}} \sum_{t=1}^{|r'_u|} \log(P_{\theta}(r'_{u,t} | p'_u, r'_{u,<t}))$$

在这个过程中，我们用 $r'_{u,t}$ 表示生成的第 $t$ 个推荐理由的令牌 $r'_{u,<t}$ 指代前 $t$ 个令牌。为优化资源，我们采用LoRA进行轻量级微调，只训练部分参数。实验结果显示，尽管使用了少量样本和小型模型，生成的理由质量与大模型相当。学生模型模仿教师模型，通过逻辑推理理解用户需求，然后提供匹配游戏类型，最后给出具体建议。这表明，通过以推理理由为标签而非直接利用LLMs生成的标签进行训练，我们赋予了小模型类似大模型的推理功能，从而提升了推荐系统的效能。

### Empowering Recommender with Reasoning Knowledge

1. ID结合方式：我们利用推理理由作为额外信息，与基于唯一标识符（ID）的传统推荐方法融合，优化封闭式学习过程，提高推荐准确性。
2. ID无关策略：我们分别将用户行为背后的推理逻辑和候选商品描述转化为用户和商品的语义表示，通过文本相似度进行推荐，这利用了小模型的推理能力，同时资源使用更为高效。

### Encoding Recommendation Rationales

通过将用户行为序列和项目 $f_i$ 的嵌入计算为语义向量，如 $\text{Vec}(f_i)$ 和 $\text{Vec}(S_u)$ ，我们能够量化它们之间的相似性。这样的处理允许我们依据生成的推理理由和项目描述来进行推荐决策，基于这些语义信息\*而非单纯的身份标识。

$$\begin{aligned} z_i^{\text{text}} &= \text{TextEncoder}(f_i), \\ s_u^{\text{text}} &= \text{TextEncoder}(r_u), \end{aligned}$$

$z_i^{\text{text}}$ 和 $s_u^{\text{text}}$ 分别是物品描述和推荐理由的文本表示，由TextEncoder（如选择BERT）生成。这个过程考虑到了语义信息，既能捕捉到用户整体兴趣的广度（开放世界的推理知识），又能反映具体商品的细节（如特性和微观偏好）。由于LLMs的推理特性 $s_u^{\text{text}}$ 包含了多层推理内容。

### Utilizing Recommendation Rationales

我们通过整合推理理由的文本表示（ $s_u^{\text{text}}$ 和 $z_i^{\text{text}}$ ）来提升基于ID的推荐。设计一个信息融合层，将此有意义的文本特征与后端模型的原始嵌入（BackboneEmb）融合，形成CombinedEmb。这



$$\begin{aligned}z_i &= g_f([g_l(z_i^{\text{text}}); z_i^{\text{id}}]), \\s_u &= g_f([g_l(s_u^{\text{text}}); s_u^{\text{SeqEnc}}]),\end{aligned}$$

我们使用ReLU激活和ID嵌入器Encoder( $f_i$ )获取物品文本表示，以及由BERT生成的用户推理理由BERT( $r_u$ )。通过将两者通过ReLU处理后相加，并进行concatenate操作，建立了一个ID无关的统一表示UnifiedEmb。这样，模型能在不依赖用户ID的情况下，结合文本和推理信息进行推荐。这种方法不仅增添了推荐的知识维度，也增强了对用户行为的理解，从而实现了推荐准确性和多样性的显著提升。

$$\begin{aligned}z_i &= g_t(z_i^{\text{text}}), \\s_u &= g_t(s_u^{\text{text}}),\end{aligned}$$

在这个模型中 $g_t(\cdot)$ 用线性层来构建，它整合了用户推理理由 $s_u^{\text{text}}$ 中的丰富知识。利用这种结构，模型能够根据物品的特征 $z_i$ 个性化推荐，同时以自然语言的形式给出推荐依据，提供了可理解的决策过程。

## Experimental Settings

### Datasets.

我们在[亚马逊](#)<sup>+</sup>的三个分类领域-----电子游戏（Games）、食品与美食（Food）和家居与厨房（Home）上进行了实验，相关数据集详情可在附录中查阅。

### Baselines.

我们在亚马逊评论的三个分类领域-----电子游戏（Games）、食品与美食（Food）和家居与厨房（Home）上运用了三种主流序列推荐模型作为基准，即GRU4Rec、SASRec和SRGNN。每个基线我们都进行了增强版本的尝试，如

### GRU4Rec<sup>+</sup>、SASRec<sup>+</sup>

和SRGNN<sup>+</sup>，通过将项ID和描述文本合并来增加项表示。同时，我们引入了独特的方法SLIM<sup>+</sup>，即ChatGPT特征扩展，它直接将ChatGPT生成的推理理由输入到推荐模型中，不进行知识蒸馏，称为ChatGPT Feature。所有这些扩展旨在研究不同来源（文本信息和推理理由）的知识如何提升推荐系统的效能。详细实现细节在附录中有提供。

### Evaluation Metrics

我们在附录中详细记录了评估方法，主要通过三个标准来衡量：NDCG@10、Hit Rate@10和HitRate@20。每次实验我们取5次运行的平均得分，并附带标准差。每个用户我们随机选取100个非目标商品，与目标项一起共101个，以此对推荐结果进行排序来评估模型<sup>+</sup>的表现。

### Overall Performance

### Improvement over Backbone Models in ID-based scenarios.

Methods	Games			Food			Home		
	NDCG@10	Hit @10	Hit @20	NDCG@10	Hit @10	Hit @20	NDCG@10	Hit @10	Hit @20
GRU4Rec	17.61 ± 0.18	30.87 ± 0.56	42.39 ± 0.62	9.10 ± 0.30	15.27 ± 0.58	19.51 ± 0.24	2.19 ± 0.21	4.17 ± 0.39	7.53 ± 0.64
GRU4Rec <sup>+</sup>	27.33 ± 0.53	44.06 ± 0.79	56.53 ± 1.24	17.75 ± 0.78	31.10 ± 1.09	45.01 ± 1.46	12.19 ± 1.02	26.76 ± 2.58	49.10 ± 3.82
SLIM <sup>-</sup>	27.70 ± 0.47	45.13 ± 0.56	57.70 ± 0.37	17.97 ± 0.70	31.78 ± 1.47	46.88 ± 2.10	13.59 ± 1.05	30.30 ± 2.01	55.85 ± 3.55
SLIM	<b>28.37 ± 0.41</b>	<b>45.68 ± 0.53</b>	<b>58.09 ± 0.58</b>	<b>18.32 ± 0.53</b>	<b>32.56 ± 1.30</b>	<b>46.92 ± 1.82</b>	<b>15.64 ± 0.51</b>	<b>34.33 ± 1.53</b>	<b>62.93 ± 3.46</b>
Improv.	3.81%	3.68%	2.76%	3.21%	4.69%	4.24%	28.3%	28.29%	28.17%
SASRec	22.73 ± 0.28	37.77 ± 0.52	51.53 ± 0.39	26.78 ± 0.24	35.78 ± 0.36	43.32 ± 0.55	2.66 ± 0.22	5.56 ± 0.72	14.93 ± 1.53
SASRec <sup>+</sup>	27.46 ± 0.19	44.88 ± 0.63	58.90 ± 0.38	30.95 ± 0.38	44.98 ± 0.53	55.61 ± 1.12	5.58 ± 0.10	11.09 ± 0.16	20.69 ± 0.77
SLIM <sup>-</sup>	<b>31.58 ± 0.35</b>	50.83 ± 0.62	63.45 ± 0.71	32.65 ± 0.15	48.01 ± 0.48	59.25 ± 0.56	5.95 ± 0.32	11.83 ± 0.62	<b>22.43 ± 0.54</b>
SLIM	31.43 ± 0.39	<b>51.11 ± 0.82</b>	<b>64.10 ± 0.26</b>	<b>32.80 ± 0.40</b>	<b>48.27 ± 0.64</b>	<b>59.30 ± 0.89</b>	<b>6.01 ± 0.19</b>	<b>12.01 ± 0.38</b>	22.29 ± 0.85
Improv.	14.46%	13.88%	8.83%	5.98%	7.31%	6.64%	7.71%	8.3%	7.73%
SRGNN	16.45 ± 0.22	29.29 ± 0.14	40.99 ± 0.52	10.99 ± 2.07	20.32 ± 4.30	32.14 ± 6.55	5.04 ± 0.83	13.48 ± 2.23	37.22 ± 3.85
SRGNN <sup>+</sup>	21.54 ± 0.64	36.77 ± 1.05	49.11 ± 1.54	11.91 ± 0.71	21.39 ± 1.91	33.63 ± 3.41	11.61 ± 1.14	25.22 ± 2.46	43.85 ± 3.58
SLIM <sup>-</sup>	22.35 ± 1.48	37.69 ± 1.53	51.29 ± 0.59	<b>12.92 ± 0.78</b>	<b>23.80 ± 1.60</b>	<b>37.22 ± 2.75</b>	11.25 ± 1.42	24.28 ± 3.19	44.05 ± 5.97
SLIM	<b>23.77 ± 0.20</b>	<b>39.81 ± 0.52</b>	<b>52.34 ± 0.63</b>	12.38 ± 0.51	22.98 ± 1.30	36.44 ± 1.72	<b>12.65 ± 1.15</b>	<b>24.51 ± 1.71</b>	<b>47.11 ± 1.61</b>
Improv.	10.35%	8.27%	6.58%	3.95%	7.43%	8.36%	5.86%	5.11%	7.21%

(1) 我们发现，无论在哪个数据集上，SLIM模型都能超越其他后端和特征扩展，这证实其能显著提升传统推荐的效能。特别是在Home数据集上，SLIM相对于GRU4Rec<sup>+</sup>，Hit Rate\@10的提升高达28.17%。这种提升归功于从ChatGPT提取的有意义推理理由，它们提供了丰富的知识，作为辅助推荐的强大信号。

(2) 意外的是，多数情况下 (22/27)，即使面对ChatGPT的特征扩展，SLIM (SLIM<sup>-</sup>) 的表现也更优。当以GRU4Rec作为后端时，它在Home数据集上的提升达到了12.68%。尽管SLIM的启发来自开源的ChatGPT，但对闭源模型的局限性，它能避免生成与推荐无关的多余信息，这表明小模型通过蒸馏聚焦于推荐相关的信息，尽管模型规模小，但推荐效果更好。

Performance in ID-agnostic scenarios.

我们探究了SLIM在不依赖具体项ID的纯序列匹配 (ID无关) 条件下的表现，即仅基于等式(1)中的CoT序列嵌入和文本项目嵌入的交互方式，我们称此为文本匹配。

Table 2: ID-agnostic Text Matching model. Comparison of recommendation performance without relying on any backbone models. SLIM-Step*i* indicates only using the *i*-th step rationales generated by SLIM.

Methods	Games			Food			Home		
	NDCG@10	Hit @10	Hit @20	NDCG@10	Hit @10	Hit @20	NDCG@10	Hit @10	Hit @20
SLIM-Step1	13.78 ± 0.59	26.08 ± 0.91	41.71 ± 0.62	13.62 ± 0.22	24.90 ± 0.59	38.15 ± 0.69	4.25 ± 0.06	9.53 ± 0.27	18.91 ± 0.77
SLIM-Step2	16.78 ± 0.66	30.09 ± 0.89	45.75 ± 0.59	13.71 ± 0.48	24.23 ± 0.78	36.89 ± 0.77	4.75 ± 0.29	10.30 ± 0.55	19.99 ± 0.61
SLIM-Step3	20.20 ± 0.31	35.57 ± 0.54	50.04 ± 0.47	15.69 ± 0.26	26.69 ± 0.68	39.19 ± 0.98	<b>4.83 ± 0.27</b>	<b>10.39 ± 0.38</b>	<b>21.05 ± 0.43</b>
SLIM <sup>-</sup>	21.75 ± 0.58	38.05 ± 0.88	53.73 ± 0.79	<b>19.08 ± 0.71</b>	<b>32.34 ± 0.74</b>	<b>44.63 ± 0.90</b>	4.75 ± 0.28	10.15 ± 0.62	19.67 ± 0.59
SLIM	<b>21.99 ± 0.22</b>	<b>38.33 ± 0.32</b>	53.59 ± 0.72	18.13 ± 0.5	30.84 ± 0.54	44.04 ± 0.66	4.49 ± 0.24	9.96 ± 0.47	19.67 ± 0.59

Merits of SLIM

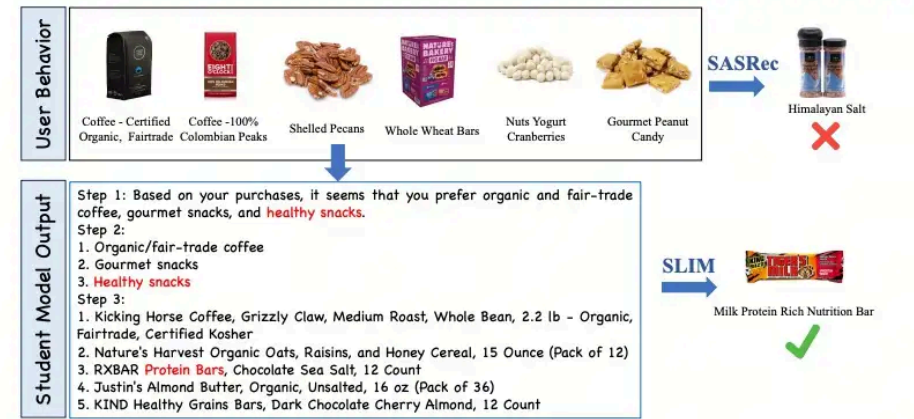


Figure 4: Case study. Comparison of the predictions for the next item obtained from SASRec and SLIM.

推荐结果的可解释性：SLIM在成功推荐目标项的同时，如图所示，SASRec则未能做到。一个例子显示，对于罕见但特征明显 (如‘健康零食’类别) 的长尾项目，SLIM能准确捕捉用户偏好，如推荐蛋白棒。SLIM通过整合序列和项目信息<sup>+</sup>，其解释清晰易懂，如Step 1和Step 2的理由，增强了推荐的有效性。

较高的G1组，改进尤为显著。这证实了SLIM在处理序列推荐的稀疏数据时具有稳定和可靠的性能。

SLIM在对抗流行度偏见方面表现出色，这是推荐系统中的一个重要问题。通过分析训练数据和推荐结果中各类商品的频率，我们发现（[补充材料](#)<sup>+</sup>提供具体图表），SLIM在推荐长尾商品时显著超越了SASRec，这个传统模型倾向于过度推荐热门商品。实验确认，SLIM有效地降低了流行度偏见，改善了长尾商品的真实可见度，有利于推荐环境的平衡。

(1) 使用LLM直接作为排序器：LLM被用来直接生成推荐列表，不通过传统的模型，而是以它自身的预测能力对物品进行排序。(2) LLM作为知识增强器：LLM作为辅助工具，通过提供额外的推理知识来增强推荐系统，它可能通过理解用户行为和物品描述，帮助优化推荐决策。

原文《Can Small Language Models be Good Reasoners for Sequential Recommendation?》

发布于 2024-05-09 11:27 · IP 属地北京

[阿里巴巴集团](#) [LLM（大型语言模型）](#) [序列推荐](#)

▲ 赞同 9 ▼ ● 添加评论 ↻ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 …



理性发言，友善互动

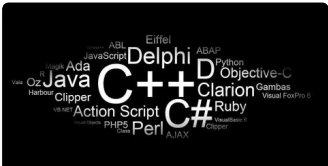


发布



还没有评论，发表第一个评论吧

推荐阅读



泛编程语言学习知识结构：  
JavaScript、Java、Python...

王下邀月熊

2023年十大编程语言对比（优缺点及应用场景）

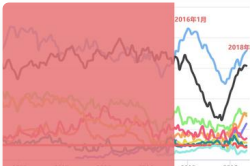
这是TIOBE上排名前十的编程语言：TIOBE是什么？TIOBE是一个编程语言流行度指数，全称为TIOBE Index（TIOBE 编程社区指数），TIOBE Index 通过对网络搜索引擎中涉及编程语言的查询结...

Jerry



帮你提升Python的27种编程语言

汇智动力IT学校



整理了5个最常见的编程  
行榜、排名逻辑&最新榜

蓝桥云课

发表于编