

# RecRanker: 指令调优LLM用于 top-k 推荐排序

原创 方方 方方的算法花园 2024年11月13日 08:56 北京

## 0 ▶ 论文概况

### 1. 论文名称:

RecRanker: Instruction Tuning Large Language Model asRanker for Top-k Recommendation

《RecRanker: 将指令微调LLM用作排序器以实现前 k 个推荐》

### 2. 论文链接: <https://arxiv.org/pdf/2312.16018>

3. 论文作者所在机构: 香港城市大学、中国科学院香港创新研究院、哈尔滨工业大学、香港中文大学、夏威夷大学马诺阿分校、商汤科技研究院、杭州电子科技大学

4. 一句话概括: 该论文提出 RecRanker, 通过多种策略 (如自适应用户采样、改进提示等) 构建指令调优数据集来训练LLM, 使其成为 top-k 推荐的排序器, 并采用混合排序方法提升模型性能, 经实验验证了其在不同推荐场景中的有效性及相关组件和参数的影响。

## 1 ▶ 挑战

1. 训练数据质量问题: 当前方法倾向于使用随机采样的低质量数据来训练 LLM, 忽视了高质量训练数据采样的重要性, 这可能无法充分挖掘 LLM 的潜力, 导致模型性能未达最优。

2. 缺乏与传统推荐模型信号的整合: 现有的文本类方法主要依赖用户和物品的文本信息, 缺乏与传统推荐模型信号的融合, 限制了推荐方法的有效性。传统推荐模型 (如 LightGCN) 能提取高阶协同信号, 对理解用户偏好至关重要, 缺少这些信息会使基于 LLM 的推荐效果受限。

3. LLM 在推荐任务中的不一致性: LLM 在推荐过程中对不同排序任务可能产生不一致的响应, 这会导致结果不可靠。此外, 单一排序方法存在局限性, 不同排序方法在不同推荐场景中的性能差异大, 且 LLM 本身的随机性机制或提示的变化也可能导致其产生不一致结果, 影响模型整体性能。

4. 计算资源需求与效率问题: LLM 在训练和推理过程中对计算资源的需求极大, 例如在实验中, 训练 LLaMA - 2 模型耗时久 (7B 模型约 4.6 小时, 13B 模型约 5.3 小时), 推理时单个指令计算时间长 (约 0.059 秒 / 项)。这使得将其应用于大规模工业推荐系统面临挑战, 如对于拥有数百万用户的系统而言, 当前基于 LLM 的推荐系统实用性较低。

## 2 ▶ 论文贡献点

1. 提出RecRanker框架: 应用指令调优的LLM执行top - k推荐中的多样化排序任务, 通过自适应用户采样构建高质量指令调优数据集, 使用改进的提示策略提升模型对用户偏好的理解与推理能力, 并采用混合排序方法提升模型性能。

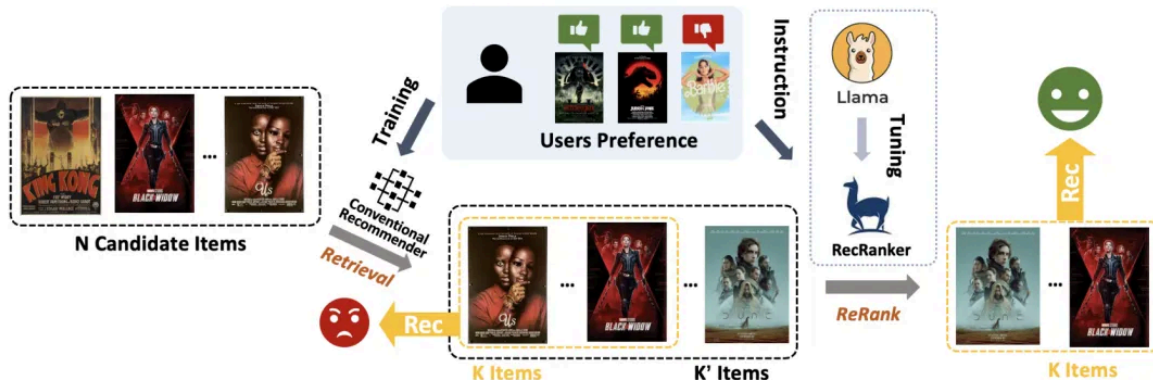


Fig. 1. An example that illustrates the application of RecRanker for the top-k recommendation scenario.

**2. 验证模型有效性：** 在三个真实世界数据集上进行广泛实验，证明RecRanker在多数情况下显著优于骨干模型，有力验证了其在推荐任务中的有效性。

### 3. 创新排序方法与策略

(1) **提出混合排序方法**，结合多种排序任务（pointwise, pairwise, and listwise），通过分配效用分数并线性插值的方式，整合不同排序任务的优势，提升推荐的全面性和有效性。

(2) **设计自适应用户采样方法**，包含重要性感知采样、聚类采样和重复采样惩罚策略，提高采样用户的质量、代表性和多样性，为模型训练提供优质数据基础。

(3) **采用位置偏移策略减轻LLM中的位置偏差，增强模型输出的一致性和可靠性**；同时提出提示增强方法，将传统推荐模型信号融入提示，使LLM能结合多种信息进行更准确的排序。

### 4. 深入研究与分析

(1) 通过消融实验，明确RecRanker各主要组件（如自适应用户采样、位置偏移、提示增强）对推荐质量的重要贡献，证明每个组件对整体性能提升的关键作用。

(2) 进行超参数研究，分析超参数对模型性能的影响，为优化模型提供指导；同时研究模型缩放（如比较7B和13B模型）和数据缩放（不同指令数量）对性能的影响，为实际应用中模型和数据的选择提供参考。

(3) 对比RecRanker与GPT模型，表明指令调优后的RecRanker在推荐任务中表现出色，强调了指令调优对通用LLM在特定推荐任务中应用的重要性。

## 3 ▶ RecRanker

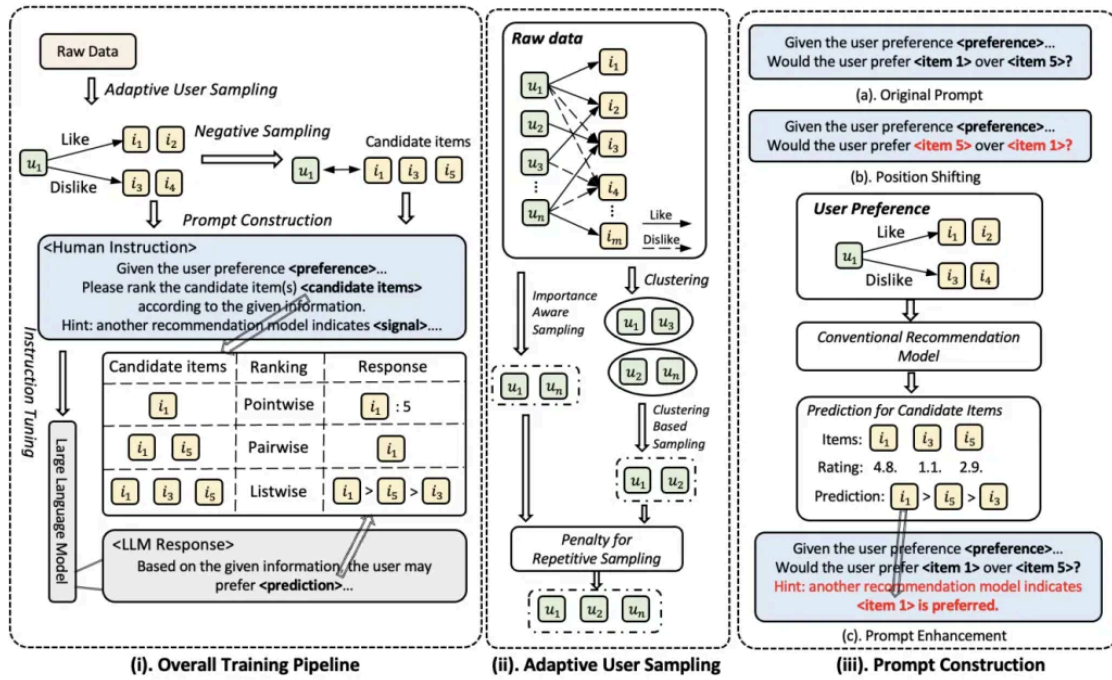


Fig. 2. (i). The overall training pipeline of RecRanker. (ii). Adaptive user sampling module, where we propose importance-aware sampling, clustering-based, and penalty for repetitive sampling to sample users. For each sampled user, corresponding candidate items are randomly selected from the items the user liked, disliked, and had no interaction with. (iii). Prompt construction, where we incorporate position shifting and prompt enhancement strategies to enhance the model performance.

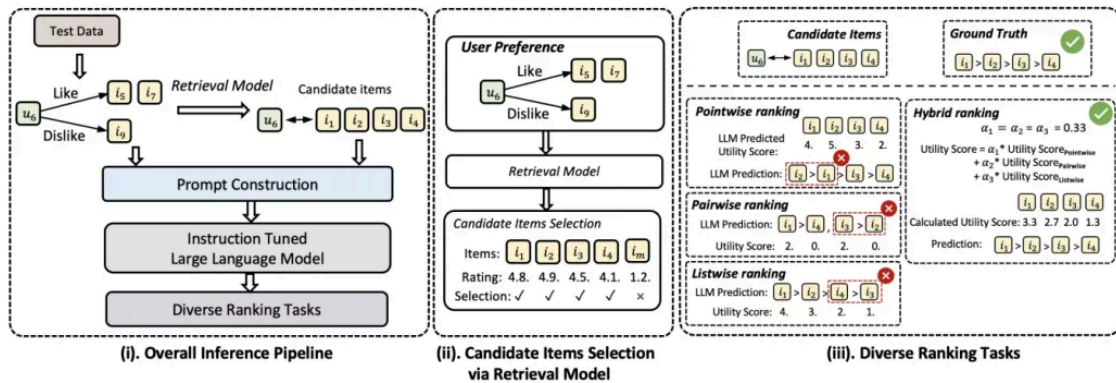


Fig. 3. (i). The overall inference pipeline of RecRanker. (ii). Candidate item selection via retrieval model, where we adopt the retrieval model to calculate the score for each item and select the highest ones as the candidate items. (iii). Comparison of the proposed hybrid ranking method with three ranking tasks during the inference stage.

## 整体结构

**1. 训练阶段主要包含四个步骤：自适应用户采样、通过负采样选择候选项目、提示构建以及指令调优。**自适应用户采样旨在获取高质量、具有代表性和多样性的用户，它融合了三种采样策略，即重要性感知采样、基于聚类的采样和对重复采样的惩罚。对于每个被采样的用户，候选项目包括用户喜欢和不喜欢的项目，以及通过常用负采样方法选择的一些未交互项目。在获取采样用户和选定项目后，为每个排序任务构建提示，并结合传统推荐模型的信号来增强提示，从而使系统能整合传统推荐系统和文本数据的优势，提升整体性能。最后，使用构建的数据通过指令调优对大语言模型（LLM）进行微调。

**2. 推理阶段**，对于测试数据中的用户，首先通过检索模型选择候选项目（此过程与训练阶段的负采样不同）。接着，按照与训练阶段相同的方法构建提示，然后由经过指令

调优的 LLM 执行各种排序任务。最后，采用混合排序方法（通过集成多个排序任务实现）来提高模型性能。

### 自适应用户采样

为构建微调数据集 *Dins*，需要从原始推荐数据集中采样用户。由于原始用户集存在数据质量和多样性问题，因此采用多集 *uins* 来表示采样用户列表。为获取高质量、代表性和多样化的数据集，引入了三种采样策略：

**(1) 重要性感知采样：**推荐场景中的数据常呈长尾分布，为优化数据质量，提出该策略。根据用户交互次数的自然对数定义用户重要性，交互次数多的用户被认为更重要，其被采样的概率与重要性成正比。这样优先选择交互次数多的用户，同时避免高交互用户过度主导数据集，确保整个用户群体都有代表性。

**(2) 聚类采样：**基于用户兴趣多样化的特点，采用聚类方法采样代表性用户。先使用检索模型（如 LightGCN）将用户表示为嵌入向量，再用 K - 均值算法将用户聚类为组，然后从每个簇中按比例采样用户。该策略保留了簇内多样性，使较大簇（代表更普遍兴趣）在样本中有更多代表，从而全面体现不同用户偏好。

**(3) 重复采样惩罚：**为增强最终多集的多样性，对重要性感知和聚类采样得到的两个多集和求和得到后，实施重复采样惩罚。为每个用户分配一个随出现次数递减的惩罚权重，出现次数多的用户被选中的概率降低，从而减少优势用户或项目的过度代表，确保不同频率用户得到更公平的采样机会。

### 候选项目选择

训练和推理阶段的候选项目选择方式不同。

**(1) 训练阶段采用负采样，**将用户喜欢和不喜欢的项目以及未交互项目混合作为候选集，其中未交互项目被视为负样本，基于此构建指令调优数据集的相对排名比较。

**(2) 推理阶段则使用检索模型，**根据模型计算的效用分数对所有项目排名，选择得分最高的（k）个项目作为候选项目，例如在直接推荐中常用 NCF、NGCF、LightGCN 等模型，在顺序推荐中常用 SASRec、BERT4Rec 等模型作为检索模型。

### prompt构建

**1.推荐系统采用多种排序任务，包括Pointwise, Pairwise, Listwise 排序，**它们在评估和排序候选项目时各有作用。为优化时间复杂度，对Pairwise排序进行改进，通过随机采样对（一对包括前位置和后位置的项目）来减少比较次数。

Table 2. Illustrative examples of instructions for three ranking tasks. For better readability, a modified version of the actual instructions employed in our experiments is shown here.

Type	Instructions	LLM Response
Pointwise Ranking	The historical interactions of a user include: <historical interactions>. How would the user rate <candidate item>?	5.
Pairwise Ranking	The historical interactions of a user include: <historical interactions>. Would the user prefer <candidate item 1> over <candidate item 2>?	Yes.
Listwise Ranking	The historical interactions of a user include: <historical interactions>. How would the user rank the <candidate item list>?	<candidate item 1>,...,<candidate item 2>



**2. 位置偏移策略用于减轻 LLM 中的位置偏差。**在训练和推理阶段，随机改变候选项目和用户偏好项目的顺序，使模型不受项目位置影响，确保其响应基于真实相关性，提高推理过程的可靠性。

**3. 现有基于 LLM 的方法常忽略传统推荐模型信号，为弥补此不足，提出提示增强方法。**将传统推荐模型（如 MF 用于点式排名，LightGCN 用于成对和列表式排名）的预测结果转换为自然语言描述并融入提示，使 LLM 能结合传统模型的协同或顺序信息，丰富排序任务的信息基础，提升排序准确性。

### 通过指令调优进行优化

构建数据集后，采用基于标准交叉熵损失的监督方式对 LLM 进行指令调优，使模型输出更符合用户意图和偏好。训练过程中，使用由自然语言指令输入 - 输出对组成的训练集来优化模型参数，通过最小化交叉熵损失，使模型在处理推荐任务时能更好地捕捉和解释用户偏好，从而提供更贴合用户个性化需求的推荐。

### 混合排序

基于单一排序方法的 LLM 推荐存在局限性，不同排序方法性能差异大且 LLM 可能产生不一致结果。受 LLM 自一致性概念启发，提出混合排序方法。

**Uhybrid**混合排名的效用分数是通过对三种不同排序任务（点式、成对、列表式）的效用分数进行线性插值计算得出的，计算公式为：

$$U_{hybrid} = \alpha_1 U_{pointwise} + \alpha_2 U_{pairwise} + \alpha_3 U_{listwise}$$

其中， $\alpha$ 是加权系数，且 $\alpha_1 + \alpha_2 + \alpha_3 = 1$ 。这些系数的值决定了混合排序对各个单独排序方法的侧重程度，从而提供了在推荐方法上的灵活性。

- **点式排序  $U_{pointwise}$** ：其效用分数最初由LLM预测的相关性分数确定，然后再加上来自检索模型的额外效用分数进行优化。

$$U_{pointwise} = U_{retrieval} + P$$

- **成对排序  $U_{pairwise}$** ：对于LLM更偏好的项目，分配一个常数效用分数。

$$U_{pairwise} = C_2$$

- **列表式排序  $U_{listwise}$** ：使用公式  $U_{listwise} = -m' \cdot C_3$

对每个项目进行评分，其中 $m'$ 表示LLM预测的项目位置（取值范围从1到 $k'$ ）， $C_3$ 是常数。

通过这种方式，混合排序方法能够整合不同排序任务的优势，综合考虑了项目在不同排序视角下的表现，从而为每个项目提供一个更全面、综合的效用分数，以实现更有效的推荐。

END

LLM与推荐 · 目录

上一篇

LLM4SBR 轻量框架：实现对话式推荐中 LLM  
的整合与工业级部署

下一篇

Meta生成式推荐器：基于万亿参数顺序变换  
器的推荐系统