【LLM论文阅读】LlamaRec:具有高效检索与排序的两阶段推荐框架

原创 方方 方方的算法花园 2024年10月17日 11:41 北京

论文标题:
LlamaRec: Two-Stage Recommendation using Large Language Models for Ranking
论文链接：https://arxiv.org/pdf/2311.02089
论文作者所在机构：伊利诺伊大学厄巴纳-香槟分校、NVIDIA
Github链接：https://github.com/Yueeeeeeee/LlamaRec
一句话概括：提出了一个基于LLM的两阶段推荐框架 LlamaRec，包括检索和排序两个阶段，采用基于 ID 的顺序推荐器进行检索，设计了基于 LLM 的排序器，通过实验证明了其在推荐性能和效率上的优势。

创新点

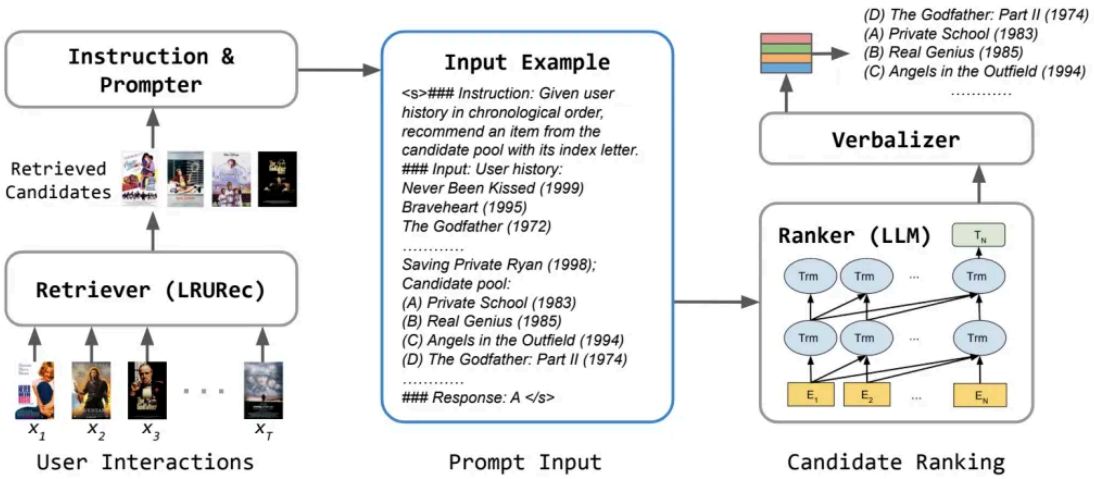
1. 提出新框架LlamaRec
提出了基于 LLM 的两阶段推荐框架 LlamaRec，包含检索和排序两个阶段，为推荐任务提供了完整的解决方案。

2.高效检索和排序方法

- 采用基于 ID 的顺序推荐器作为检索器（如 LRURec），能高效生成候选项目，不受用户历史长度影响。
- 设计了基于 LLM 的排序方法，通过精心设计的模板将用户历史和候选项目转换为文本，对预训练的 LLM 进行参数高效微调（PEFT），并采用 verbalizer 将 LLM 头部输出转换为候选项目的概率分布，无需额外参数，显著提高了基于 LLM 的推荐的时间和内存效率。

3.实验验证有效
在基准数据集上验证了 LlamaRec 的有效性，实验结果表明 LlamaRec 在推荐性能和效率上均优于基线方法，在顺序推荐任务中取得了显著的改进。

具体方案



检索阶段

（1）模型选择：采用线性递归单元的LRURec作为检索模型$f_{retriever}$，它是一个小规模的顺序推荐器，利用线性递归单元高效处理输入序列。

（2）训练与优化：通过自回归训练进行优化，以捕获用户转换模式并生成预测的项目特征。

（3）推理计算：在推理时，计算项目嵌入和预测特征之间的点积作为项目分数，为每个输入序列收集LRURec的前k（实验中k = 20）个推荐，这些候选项目将用于下一阶段的排序。

排序阶段

（1）LLM选择：选择Llama 2的7B版本作为排序模型f_{ranker}。

（2）文本输入构建：构造文本输入时，先添加一个描述任务的指令，然后是用户历史和候选项目的标题。具体的prompt模板为：

> **### Instruction**: Given user history in chronological order, recommend an item from the candidate pool with its index letter.
> ### Input: User history: *{ history }*;
> Candidate pool: *{ candidates }*
> **### Response**: *{ label }*

其中history、candidates和label分别由历史项目标题、候选项目标题和每个数据示例的标签项目替换，在推理时，标签位置留空用于预测。

（3）排序得分计算

verbalizer 是一种将LLM头部输出转换为候选项目排名分数的工具。在 LlamaRec 框架中，它起到了关键作用，使得模型能够在不进行长文本生成的情况下，高效地对候选项目进行排序。

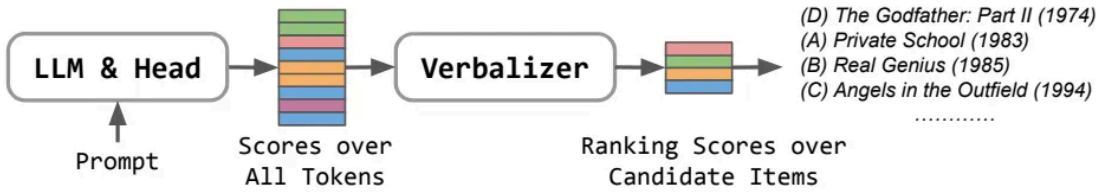


Figure 2: Verbalizer in our LlamaRec ranker.

具体计算过程：

- **候选项目标识**：首先，利用索引字母来识别候选项目。例如，对于电影推荐，候选项目可能是 (A) Private School (1983)、(B) Real Genius (1985)、(C) Angels in the Outfield (1994) 等

- **映射真实项目**：将真实项目（ground truth item）映射到相应的索引字母。这样，每个候选项目都有了一个对应的索引token。
- **计算候选分数**：通过从 LLM 头部检索索引字母的对数（logits）来计算候选分数。这些检索到的分数对应于索引字母内的下一个token概率分布。也就是说，LLM 对每个索引字母出现的可能性给出了一个分数，这个分数就作为候选项目的得分。

训练过程中的处理：

在训练时，将真实项目的索引字母作为标签（label）。通过这种方式，最大化真实项目的得分，使得模型学习到根据用户历史和候选项目来正确地对项目进行排序。这种训练范式与 LLMs 通常采用的下一个标记预测任务是相同的。

（4）训练优化

在LlamaRec实现中，对提示的响应部分应用指令调整并优化模型，即仅计算每个数据示例中prompt的标签标记（即索引字母和EOS token）的损失，因为优化整个输入不会带来进一步的改进，而将损失计算限制在标签部分在训练中更有效。

实验结果

数据集和基线方法

- 在 ML - 100k、Beauty 和 Games 三个数据集上进行评估，采用了多种先进的顺序推荐器作为基线方法，包括 RNN 模型（NARM）、基于transformer的推荐器（SASRec、BERT4Rec）和基于线性递归的 LRURec，以及基于 LLM 的顺序推荐器（P5、PALR、GPT4Rec、RecMind 和 POD）。
- 实验遵循留一法策略，使用平均倒数排名（MRR@k）、归一化折损累计增益（NDCG@k）和召回率（Recall@k）作为评估指标，k 取值为 5 和 10。

主要结果

- LlamaRec 在所有数据集的所有指标上均表现最佳，与表现最佳的基线方法相比，在 ML - 1M、Beauty 和 Games 上分别实现了 11.99%、3.99% 和 11.06% 的平均改进。在 ML - 100k 上性能提升最大，MRR@5、NDCG@5 和 Recall@5 分别提高了 12.82%、16.02% 和 20.85%。
- 在有效检索子集中，LlamaRec 的性能提升更为显著。与基于 LLM 的基线方法相比，在 Beauty 数据集上平均性能提升 14.31%。

效率比较

评估了 verbalizer 方法与生成方法的排名效率，结果表明 LlamaRec 仅需一次前向传递即可获得所有候选项目的排名分数，而生成方法在平均标题长度为 20 时，推理时间为 56.16s，LlamaRec 的推理时间始终在 1s 以内，效率大幅优于基线生成方法。

LLM与推荐 15　LLM论文阅读 13　#LLM学习 12

LLM与推荐 · 目录

上一篇
【LLM论文阅读】LLMRanker：利用LLM对候选物品集合进行zero-shot排序

下一篇
【LLM论文阅读】利用 LLMs 进行推荐多样性重排序