Collaborative Large Language Model for Recommender Systems

Yaochen Zhu*
University of Virginia
uqp4qh@virginia.edu

Liang Wu
LinkedIn Inc.
liawu@linkedin.com

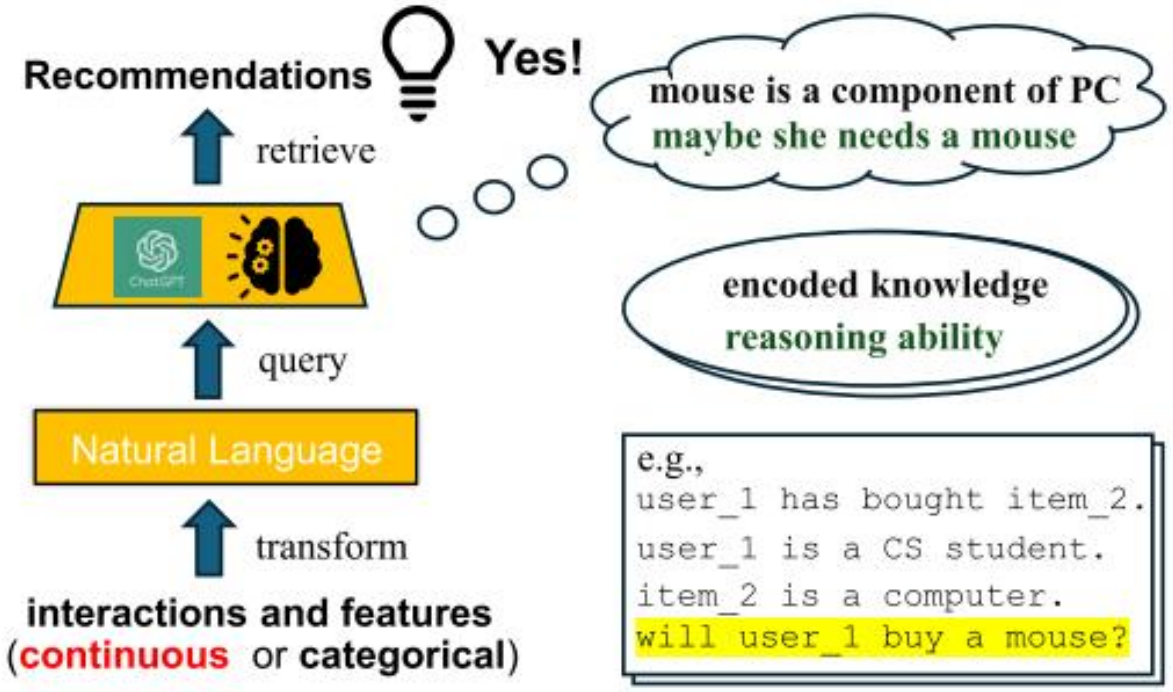
Qi Guo
LinkedIn Inc.
qguo@linkedin.com

Liangjie Hong
LinkedIn Inc.
liahong@linkedin.com

Jundong Li
University of Virginia
jundong@virginia.edu

WWW24

引言——背景



早期LLM应用于推荐系统的基本流程：

步骤一：将用户/物品表示转化为自然语言，能够被语言模型理解。
相关信息包括交互记录、特征、候选项目（连续数据or分类数据）（e.g. 评分、品牌）

步骤二：利用提示查询LLM生成推荐
将转换好的提示作为查询输入LLM。最后根据LLM的输出从候选池中检索具体的item。

（可选）微调
如果有真实的数据，预训练的LLM可以在真实交互数据和特征数据上进行微调。（e.g. 购买记录、用户主页 ）

引言——研究动机

- 表示用户/项目的挑战

① 使用伪ID（如“user_4332”）表示。["user_4332", "clicked", "item_5678"]

② 使用描述性Token（如物品标题）表示用户/物品。["Alice", "clicked", "Apple iPhone 13"]

③ 真实Token方法为，每个用户和物品引入独立的、全新的Token。这些Token被直接作为模型的输入标记，且不会被进一步分词。[<user_4332>, "clicked", <item_5678>]

表示方式	优点	缺点
伪ID	简单易用，ID标识符唯一，直接表示不同用户/物品	分词后可能引入虚假相关性，缺乏语义信息
描述性Token	提供语义信息，易于直接输入LLM处理	引入强归纳偏置，可能忽略其他特征
真实Token	精确表示用户/物品，避免分词问题	导致词汇表膨胀，稀释模型能力，随机初始化的Token缺乏语义

引言——研究动机

- **模型建模的挑战**

① 但将交互历史转化为自然语言时，由于自然语言本身带有先后顺序，可能引入虚假的时间相关性。

② LLM未专门针对推荐任务设计，可能从用户/物品文本特征中捕获与推荐无关的噪声信息。

③ LLM通常通过自回归逐Token生成推荐项，效率低于传统基于ID的推荐方法。

推荐任务：为用户生成5个电影推荐。

- 自回归方式：
  - 第1步：生成第1个推荐（"Inception"）。
  - 第2步：基于第1个推荐生成第2个推荐（"Avatar"）。
  - ...重复直到生成第5个推荐。

用户交互历史：

- 用户喜欢的电影：`["Inception", "Avatar", "The Matrix"]`。

转化为文本输入：

- `"The user watched Inception, Avatar, and The Matrix."`

物品描述：

- `"This phone comes in three colors: red, blue, and green."`

用户交互：

- 用户查看了此物品，但主要关注性能而非颜色。

引言——研究动机

- 实际应用的挑战

① 为避免生成不属于候选池的推荐项（幻觉问题），伪ID或描述性方法需要显式提供候选池，这对大规模推荐场景不现实。

② 在候选池较大且需要实时响应的场景中，LLM的生成效率和延迟表现仍然难以满足实际需求。

- **候选商品池**（库存）：
  `["iPhone 14", "Samsung Galaxy S23", "Google Pixel 7", "Xiaomi 13"]`
- **用户需求**：推荐一款手机。
- **未显式提供候选池的提示**：
  LLM提示为：
  `"The user is looking for a smartphone. What would you recommend?"`
- **幻觉问题**：
  LLM可能生成：
  `"We recommend the OnePlus 12"`（此商品不在库存中）。

应用场景：电商平台的实时推荐。

- 候选池：10万种商品。
- 用户在页面点击后，推荐系统需要在50毫秒内生成10个推荐项。

方法—User/Item Token的扩展

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **词汇扩展**

将用户和物品ID作为新的Token加入到LLM的词汇表中，例如<user_i>和<item_j>。
这些Token不会被进一步分解（例如不会将<user_1234>分成“user”、“1234”）。

- **Token嵌入**

此时，模型的语料库有了两种token，自然语言token和ID token
为了让LLM理解新引入的用户/物品Token，需要将其映射为嵌入向量。

面临问题：直接对包含用户/物品ID和自然语言词汇的异构序列进行语言建模，会因为新引入的ID Token稀释了自然语言Token（如GPT的50k词汇）而导致效果不佳

方法—相互正则化预训练

解决方法：

为ID token扩展两套嵌入、为基础模型添加两个预测头、为预训练构建两种语料

分别捕捉交互语义（用户 - 物品交互）和内容语义（用户 / 物品文本特征）

协作嵌入：采样方式：服从零均值高斯分布
作用：捕捉用户 / 物品在交互历史中的交互语义（如用户偏好、物品流行度）

内容嵌入：采样方式：以协作嵌入为中心的条件高斯分布
作用：捕捉用户 / 物品的文本特征语义（如用户简介、物品描述）

方法—相互正则化预训练:协作LLM和内容LLM

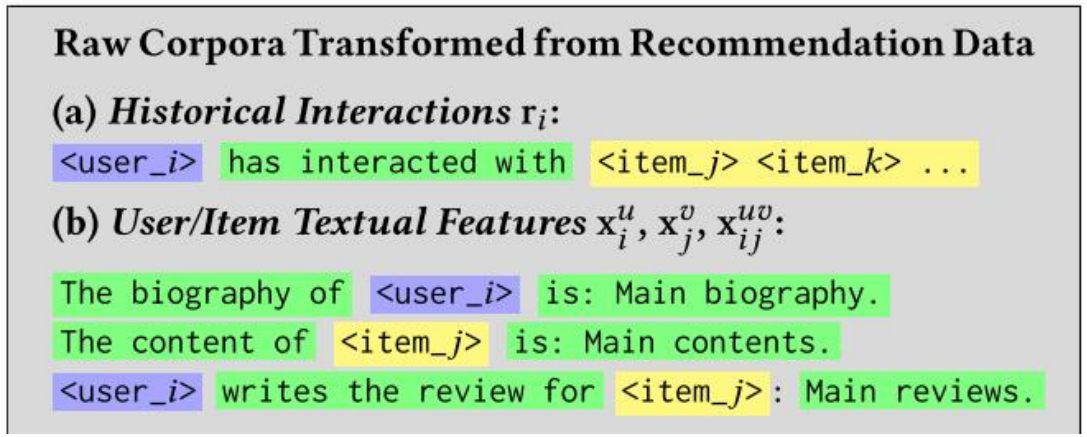
基础模型：
作用：CLLM4Rec 基础模型（Decoder-based）可以将输入token序列进行嵌入。输入的token是维度为 N+I+J的独热向量，N代表词汇表的大小，也就是模型能识别的普通词汇的数量；I是用户的数量；J是物品的数量。

固定与可训练部分：对于 CLLM4Rec 基础模型 ，只有用户 / 物品token嵌入是可训练的，而自然语言词汇嵌入以及 LLM 骨干网络的其他部分都保持固定。

协作LLM（Collaborative LLM）（生成下一个item）：
通过给基础模型添加项目预测头，以预测用户可能交互的下一个项目。

内容LLM（Content LLM）（生成下一个自然语言词汇）：
通过给基础模型添加词汇预测头，以预测文本中的下一个词汇。

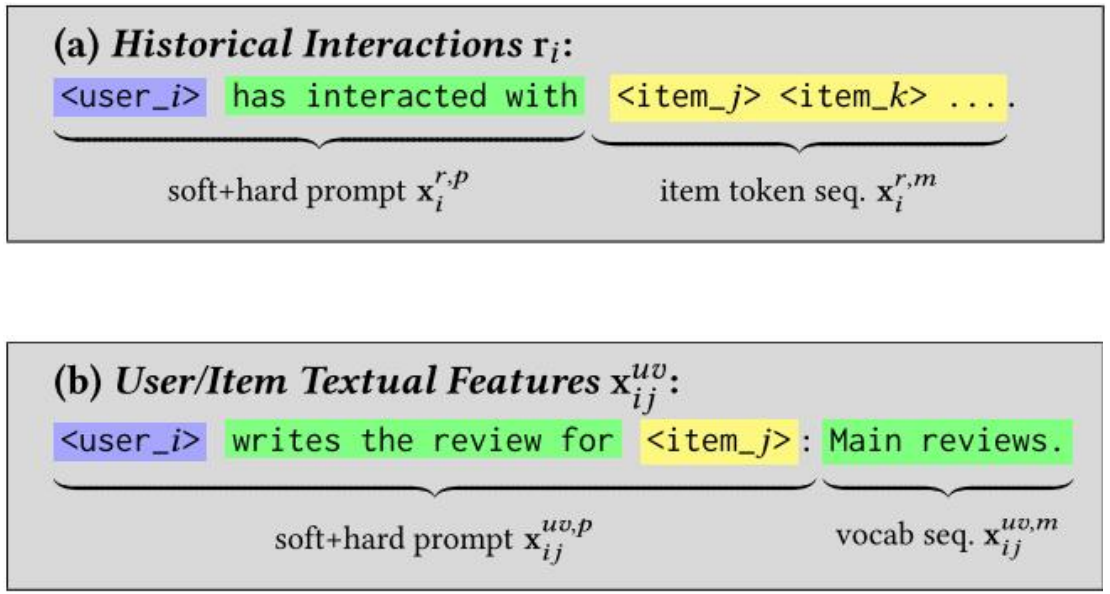
方法—相互正则化预训练：语料库



将用户-物品交互数据和文本特征的文档格式主要分为以下两类：

- 历史交互
- 用户/物品文本特征

方法—相互正则化预训练:Soft+Hard Prompting



将预训练输入文档分解为两部分：

提示词部分：
软文本（用户/物品token）+
硬文本（相关自然语言提示词）

主文本部分：
训练协同模型时（item list）
训练内容模型时（自然语言序列）

方法—相互正则化预训练：相互正则化过程

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **挑战：**

内容 LLM（处理文本）：直接优化文本生成（如评论）会学到噪声（如“包装精美”与推荐无关），需用协作信息（用户交互历史）引导其聚焦推荐相关语义（如“性能高效”）。

协作 LLM（处理交互）：仅用稀疏交互数据训练易过拟合（如用户仅买一次的物品被误判为长期兴趣），需用文本信息（如物品描述）补充缺失的语义。

- **相互正则化策略：**

通过双向约束让两类 LLM 的嵌入（用户 / 物品的协作嵌入 vs. 内容嵌入）在潜在空间中接近，形成“交互语义”与“文本语义”的对齐。

方法—相互正则化预训练：相互正则化过程

将协作 LLM 和内容 LLM 的生成过程、用户 / 物品嵌入的依赖关系建模为一个联合概率分布：

$$p(\text{交互序列}, \text{文本序列}, \text{所有嵌入}|\text{提示}) = \underbrace{\text{协作LLM生成项}}_{\text{拟合交互数据}} \cdot \underbrace{\text{内容LLM生成项}}_{\text{拟合文本数据}} \cdot \underbrace{\text{互正则化项}}_{\text{嵌入对齐约束}} \cdot \underbrace{\text{先验项}}_{\text{弱约束，可忽略}}$$

协作 LLM 生成项：生成用户交互的物品序列（如 “用户购买了 A、B”），对应协作 LLM 的语言建模损失（预测下一个物品token）。

内容 LLM 生成项：生成用户 / 物品的文本（如评论），对应内容 LLM 的语言建模损失（预测下一个词汇token）。

互正则化项：假设内容嵌入（如用户文本嵌入）服从以协作嵌入（用户交互嵌入 ）为中心的高斯分布，强制两者在向量空间中接近（物品同理）。用户的交互历史（如购买记录）定义其核心偏好（协作嵌入），而用户的文本特征（如评论、简介）需围绕该偏好分布。例如，若用户多次购买 “游戏本”（协作嵌入），其评论中 “高性能显卡” 的语义（内容嵌入）会被强化并向协作嵌入靠拢，无关噪声（如 “包装颜色”）则被抑制。

先验项：协作嵌入的零均值先验（如 “用户嵌入应分布在原点附近”），但因互正则化约束更强，实际训练中忽略（设精度参数为 0）。

方法—相互正则化预训练 L-step/C-step交替优化

L-step 和 C-step 是 CLLM4Rec 中互正则化预训练策略的核心优化步骤，通过交替固定一类嵌入、优化另一类嵌入，实现协作 LLM 与内容 LLM 的双向约束。

***L-step*（优化协作 *LLM*：交互嵌入←文本嵌入约束）**

$$\mathcal{L}_{\text{l\_step}}^{\text{MAP}}\left(\mathbf{z}_i^{l,u}, \mathbf{Z}_i^{l,v}; \theta\right) = \sum_k \underbrace{-\ln p_{l\hat{l}m_l}^{f_l}\left(\mathbf{x}_{i,k}^{r,m} | \mathbf{x}_{i,1:k-1}^{r,m}, \mathbf{x}_i^{r,p}\right)}_{\text{LM loss for collab. LLM}}$$

$$\underbrace{+\frac{\lambda_c}{2}\left\|\mathbf{z}_i^{l,u} - \hat{\mathbf{z}}_i^{c,u}\right\|_2^2 + \sum_k \frac{\lambda_c}{2} \cdot \left\|\mathbf{z}_{ik}^{l,v} - \hat{\mathbf{z}}_{ik}^{c,v}\right\|_2^2}_{\text{MR loss with content LLM}} \underbrace{+\frac{\lambda_l}{2}\left\|\mathbf{z}_i^{l,u}\right\|_2^2 + \frac{\lambda_l}{2}\left\|\mathbf{z}_j^{l,v}\right\|_2^2}_{\text{Prior loss}} + C_l, \tag{7}$$

训练协同LLM时的损失函数：

语言模型损失（LM loss for collab. LLM），用于学习用户和项目之间的交互。

相互正则化损失（MR loss with content LLM），用于确保协作LLM和内容LLM中的嵌入一致性。

先验损失（Prior loss），用于正则化用户和项目的嵌入。防止过拟合。（影响不大，后面训练时置为0了）

方法—相互正则化预训练 L-step/C-step交替优化

L-step 和 C-step 是 CLLM4Rec 中互正则化预训练策略的核心优化步骤，通过交替固定一类嵌入、优化另一类嵌入，实现协作 LLM 与内容 LLM 的双向约束。

***C-step*（优化内容 *LLM*：文本嵌入←交互嵌入约束）**

训练内容LLM时的损失函数：

语言模型损失（LM loss for content LLM），用于学习用户和项目内容（一些文本侧信息）之间的关联。

相互正则化损失（MR loss with collab. LLM），用于确保内容LLM中的嵌入与协作LLM中的嵌入一致。

$$\mathcal{L}_{\text{c\_step}}^{\text{MAP}}\left(\mathbf{z}_i^{c,u}, \mathbf{z}_j^{c,v}; \theta\right) = \sum_k \underbrace{-\ln p_{\hat{llm}_c}^{f_c}\left(\mathbf{x}_{ij,k}^{uv,m} \middle| \mathbf{x}_{ij,1:k-1}^{uv,m}, \mathbf{x}_{ij}^{uv,p}\right)}_{\text{LM loss for content LLM}} + \underbrace{\frac{\lambda_c}{2}\left\|\mathbf{z}_i^{c,u} - \hat{\mathbf{z}}_i^{l,u}\right\|_2^2 + \frac{\lambda_c}{2} \cdot \left\|\mathbf{z}_j^{c,v} - \hat{\mathbf{z}}_j^{l,v}\right\|_2^2}_{\text{MR loss with collab. LLM}} + C_c, \tag{8}$$

方法—相互正则化预训练:随机重排

① 但将交互历史转化为自然语言时，由于自然语言本身带有先后顺序，可能引入虚假的时间相关性。

为了解决这个问题，文章提出了一种随机物品重排序策略，即在优化协同语言模型（LLM）时，固定提示文本，但是随机打乱item list token的顺序。

方法—面向推荐的微调

•功能局限：预训练后的 CLLM4Rec 只能基于提示完成物品 / 词汇令牌序列，而不能直接进行推荐，自然语言处理（NLP）和推荐系统（RS）之间的差距仍未完全消除。

•计算成本高：若简单地将协作 LLM 用作推荐模型，由于推荐物品是通过自回归顺序生成的，会导致巨大的计算成本。

- **随机掩码历史交互数据**

对于每个用户，随机掩盖一部分物品

- **生成提示**

通过掩码后的交互数据生成推荐的输入提示（prompt）。形式为 “<user_i> has interacted with <item_j’> <item_k’> the user will interact with:” 。

- **引入物品预测头**

在微调过程中，RecLLM（微调的模型）加入了一个物品预测头，用于根据用户历史交互生成推荐。该预测头通过多项分布来预测目标物品。所有被掩码的物品（用多热向量表示）作为目标。

方法—进行预测

1. 生成推荐导向的提示

当要为用户进行推荐时，首先需要将该用户的整个历史交互记录 转换为推荐导向的提示

提示中包含了用户所有交互过的物品信息。例如，如果用户 之前购买过商品 A、商品 B 和商品 C，那么提示可能是 “<user_i> has interacted with <item_A> <item_B> <item_C> the user will interact with:”

2. 输入模型进行前向传播

将生成的提示 输入到 RecLLM（微调阶段的 CLLM4Rec）模型中。RecLLM 模型通过预测头可以计算出所有物品的多项式概率这里的多项式概率表示了用户与每个物品进行交互的可能性大小。

3. 选择推荐物品

最后，从所有物品的概率分布 中，选择用户 未交互过的物品，并找出其中得分最高的前

M个物品作为推荐结果。例如，如果M=5，那么就会挑选出用户 没有购买过的商品中，预测概率最高的 5 个商品推荐给用户。

实验

•RQ1：CLLM4Rec 通过跨范式融合（ID+LLM）和预训练知识迁移，在准确性上显著优于单一范式基线。

•RQ2：互正则化（λc=1）和随机重排序是预训练阶段的核心策略，分别解决噪声捕捉和顺序依赖问题。

•RQ3：微调阶段通过掩码提示和多项预测头提升效率，嵌入迁移技术支持工业级低延迟部署。

CLLM4Rec 通过互正则化预训练融合用户 / 物品的交互 ID 与文本语义，结合任务导向微调高效生成推荐，在稀疏性、噪声场景和工业部署中展现出超越传统及 LLM-based 推荐系统的性能。

Thanks