

百度2024：解密大模型（LLM）在推荐系统的表示学习中的应用

**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

14 人赞同了该文章

Introduction

推荐系统利用深度学习和图神经网络⁺，如NGCF和LightGCN，显著提升个性化推荐。然而，它们过分依赖ID信息，忽视了文本信息，且易受隐式反馈（如点击偏见）的噪声干扰。为此，我们提出了RLMRec，通过结合LLMs（大型语言模型⁺）增强表示学习，捕捉更多深层次的语义特征。我们通过辅助文本和跨视图对齐技术处理文本信息，利用互信息优化表示，以应对数据质量和偏见问题，确保在噪声环境下推荐系统的稳健性。

- 本研究探讨如何利用LLMs强化推荐系统，通过优化语义表示与协同关系建模的融合，提升推荐效能。面临的问题包括LLMs的计算资源需求随用户数据增长而增加的可扩展性挑战，以及在有限的令牌限制下，LLMs可能错过全局行为理解和推荐的准确性问题。详细分析和实例可见附录。
- 我们提出RLMRec，这是一种模型无关的表示学习框架，它依据理论创新设计。通过对比式或生成式学习策略，我们增强了学习到的表示，以提高推荐系统的性能。
- 我们构建了理论支持，证明在表示学习中融入文本信息的有效性。我们通过最大化互信息来阐明如何利用文本信息提升表示的精确度。
- 我们将RLMRec，并进行了实证测试，验证了其有效性。此外，我们还检验了它在处理含噪声和不完整数据时的稳健性，显示了它应对实际挑战的能力。

Methodology

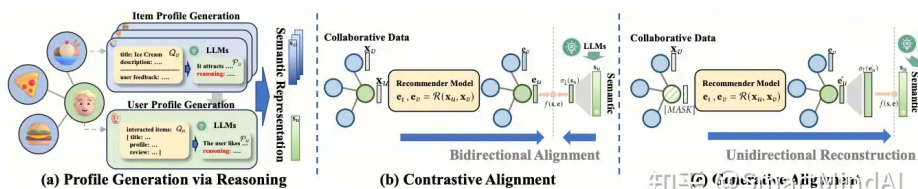


Figure 3: The overall framework of our proposed LLM-enhanced representation learning framework RLMRec.

Theoretical Basis of RLMRec

在这个表达式中，

$$P(\mathbf{e}_u, \mathbf{e}_v | \mathcal{X})$$

表示用户 u 和项目 v 在推荐系统中潜在表示 \mathbf{e}_u 和 \mathbf{e}_v 的条件概率⁺，其中 \mathcal{X} 是影响他们关系的所有相关信息（如

赞同 14

添加评论

分享

喜欢

收藏

申请转载

...

在[推荐系统](#)⁺中，这个公式常用于预测用户对未接触项目的兴趣，从而进行个性化推荐。



$$p(\mathbf{e}|\mathcal{X}) \propto p(\mathcal{X}|\mathbf{e})p(\mathbf{e})$$

在实际推荐应用中，观测到的用户-项目交互数据 \mathcal{X} 如实反映了用户行为，但常含有噪声，如非理性的流行度偏见（误点击）和未被充分揭示的兴趣（未互动的正样本）。这干扰了通过学习得到的用户和项目表示 \mathbf{e} 的准确性。

为克服这个问题，我们在方法中引入了隐含的先验信念 \mathbf{z} ，它源于对推荐系统的内在理解。这个信念帮助区分真实有效的交互和噪声，增强了模型对正样本的识别能力。因此，生成推荐表示 \mathbf{e} 的过程需要同时考虑 \mathbf{z} 作为引导，以及在学习过程中不可避免的噪声。

我们提出增强的文本用户偏好学习，以消除无关信息干扰。为了精准地捕捉用户偏好，我们引入了附加的、信息丰富的提示，如用户和项目的描述。这些描述通过语言模型转化为向量 $\mathbf{s} \in \mathbb{R}^d$ 用来理解用户偏好的深层语义。关键在于 \mathbf{s} 和 \mathbf{e} 都能提取与用户-项目交互相关的共享特征，这与我们之前提到的先验信念 \mathbf{z} 相符，强调了包含有益信息的重要性。这样做的目的是确保学习到的表示既准确又聚焦于用户的真实兴趣。

$P(\mathbf{e}|\mathbf{s}, \mathbf{z})$ 这样结合两者，我们力求更全面地了解用户喜好，从而显著提升推荐系统的精确性和效率。

$$\mathbf{e}^* = \arg \max_{\mathbf{e}} \mathbb{E}_{p(\mathbf{e}, \mathbf{s})} [p(\mathbf{z}, \mathbf{s}|\mathbf{e})].$$

优化条件概率的策略旨在保证推荐模型能从先验信念 \mathbf{z} 中获取纯净信息，同时利用语义表示 \mathbf{s} ，以提升推荐的关联性和效益。

通过这样结合，我们期望学习到的表示 \mathbf{e} 更能精准反映用户的真正需求，从而改善推荐系统的性能。定理1说明，当假设隐藏的先验信念 \mathbf{z} 已知时，最大化[后验概率](#)⁺

$$\mathbb{E}_{p(\mathbf{e}, \mathbf{s})} [p(\mathbf{z}, \mathbf{s}|\mathbf{e})]$$

等于最大化两个方面信息的联合-----用户表示 \mathbf{e} 与LLM生成的语义表示 \mathbf{s} 之间的互信息。证明中指出，由于用户和项目描述固定 $p(\mathbf{s})$ 不随训练改变，所以优化过程简化为优化 \mathbf{e} 与 \mathbf{s} 的互信息。这样做有助于模型在融合 \mathbf{z} 的直接信息和与 \mathbf{s} 共享的[语义信息](#)⁺上达到最优，从而增强推荐的关联性和效果。

$$\begin{aligned} \mathbb{E}_{p(\mathbf{e}, \mathbf{s})} [p(\mathbf{z}, \mathbf{s}|\mathbf{e})] &\propto \mathbb{E}_{p(\mathbf{e}, \mathbf{s})} \log \left[\int_{\mathbf{z}} \frac{p(\mathbf{z}, \mathbf{s}|\mathbf{e})}{p(\mathbf{s})} d\mathbf{z} \right] \\ &= \mathbb{E}_{p(\mathbf{e}, \mathbf{s})} \log \left[\frac{\int_{\mathbf{z}} p(\mathbf{z}, \mathbf{e}|\mathbf{s}) d\mathbf{z}}{p(\mathbf{e})} \right] \\ &= \mathbb{E}_{p(\mathbf{e}, \mathbf{s})} \log \left[\frac{p(\mathbf{e}|\mathbf{s})}{p(\mathbf{e})} \right] = I(\mathbf{e}, \mathbf{s}). \end{aligned}$$

根据数学表达式，我们寻求找到使联合信息 $I(\mathbf{E}; \mathbf{S})$ 最大的 \mathbf{e} ，其中 \mathbf{E} 是所有样本的联合表示 \mathbf{S} 是对应的LLM生成的语义表示。这个过程可以通过对每个样本对 $(\mathbf{e}_i, \mathbf{s}_i)$ 计算并累加单个样本的互信息 $I(\mathbf{e}_i; \mathbf{s}_i)$ 来实现，通过迭代的优化算法如梯度下降来优化。通过最大化这个总体互信息，我们不仅利用了隐含的先验信念 \mathbf{z} 提供的潜在结构信息，还减少了噪声的影响。这种方法增强了推荐系统的精确性，因为每个用户和项目的独特表示能更好地反映他们的真正兴趣。

$$\max_{\mathbf{e}} \mathcal{L}(\mathbf{e}) = \sum_{i=1}^N [\log f(\mathbf{s}_i, \mathbf{e}_i) - \log p(\mathbf{s}_i)]$$

其中

$\log f(\mathbf{s}_i, \mathbf{e}_i)$ 是基于[似然函数](#)⁺的对数，通过估计 $p(\mathbf{s}_i|\mathbf{e})$ 和 $p(\mathbf{s}_i)$ 来估计单个样本对的贡献。我们通过优化这个下界来达到目的，实质上是在追求 \mathbf{e}_i 和 \mathbf{s}_i 间潜在联系的最大化，同时削弱无关信息的干扰。这样，优化后的 \mathbf{e} 更能聚焦于用户与项目的真实关联，提升推荐系统的准确性和效果。

User/Item Profiling Paradigm

- **内容profile (Item profile)**：它应该清晰地传达出该物品倾向于吸引的具体类型用户，为用户供其特性、品质与这些用户群体偏好和兴趣相一致明确描述。这样的简介有助于确保推荐系统能够准确识别潜在目标用户，避免无关或噪声信息的干扰，提高推荐的精准度和有效性。

在某些情况下，原始数据可能包含与用户和商品相关的文本特性。例如，在Yelp数据集中，用户为访问过的商家撰写评论，商家拥有诸如位置和类别等属性。然而，这种文本数据通常携带着大量无关噪声，导致一系列问题：

在处理包含文本特性的原始数据时，Yelp和Steam等数据集面临诸如缺失属性和噪音文本的问题。这些问题源于文本中大量的无关信息。为克服这些挑战，我们提出利用大型语言模型（LLMs）的新型策略：

1. 文本去噪：利用LLMs消除无关文本，保留与用户偏好相关的内容，提升信息质量。
2. 信息融合：结合LLMs生成的摘要和直接从数据中提取的属性，形成综合的用户和商品简介。
3. 特征提取：虽主要依赖LLM生成，但仍从文本中提取关键属性，保证信息完整性。
4. 动态更新：模型能适应并随时间自动更新，以反映用户行为和环境变化。

通过这种方法，我们旨在创造更精确的用户和商品简介，以优化推荐系统的性能和准确性。

Profile Generation via Reasoning.

近年来研究证实，结合过程推理的LLMs能减少虚假内容，提升输出质量。为此，我们设计了系统提示 $S_{u/v}$ ，明确指导LLMs生成用户或商品的轮廓，包括属性和行为逻辑。这一过程包括定义输入输出内容和预期格式。生成结果中，我们强调包含推理过程，视之为输出一部分。通过结合用户轮廓生成提示 Q_u 和商品轮廓提示 Q_v ，LLMs能生成精确的个人资料。

操作步骤如下：

1. 系统提示：明确LLMs的任务，生成用户或商品详尽描述。
2. 轮廓聚焦：利用上下文提示 Q_u 和 Q_v ，引导LLMs聚焦关键信息。
3. 过程推理：嵌入推理能力，确保内容基于现实情境和逻辑。
4. 内容整合：合并LLMs输出与直接数据提取的属性，生成完整简介。
5. 动态更新：随着数据更新，模型持续学习并调整，保持简介的时效性和准确性。

通过这种方法，我们期望借助LLMs生成的高质量简介，提高推荐系统的预测精度。

$$P_u = LLMs(S_u, Q_u), \quad P_v = LLMs(S_v, Q_v)$$

Item Prompt Construction.

对于商品 v ，位于集合 \mathcal{V} 内的文本数据被划分成四部分：标题 α 、原始描述 β 、专属数据属性 γ （包含 γ_1 到 $\gamma_{|\gamma|}$ ）和用户评论 \mathbf{r} （共 n 条）。我们定义生成商品简介的输入提示 Q_v 如下：

1. 标题（Title）：明确商品基本信息，如名称或标签。
2. 原始描述（Original Description）：保留商品原始描述，展示其基本特征和品质。
3. 专属性（Dataset Attributes）：利用 γ ，提取来自数据集的特定信息，如尺寸、颜色等。
4. 用户评价（User Reviews）：整合 \mathbf{r} ，收集用户实际反馈，反映用户对商品的主观感受和需求。

这样组织输入提示，确保LLMs在生成简介时能综合考虑所有信息来源，从而提升生成内容的准确性和相关性。

$$Q_v = f_v(\mathbf{x}) \text{ w.r.t. } \mathbf{x} = \begin{cases} [\alpha, \beta], & \text{if } \beta \text{ exists,} \\ [\alpha, \gamma, \hat{\mathbf{r}} \subset \mathbf{r}], & \text{other wise.} \end{cases}$$

在我们的策略中，针对每个物品 v ，我们定义函数 $f_v(\cdot)$ 来整合多元文本特征。当原始描述 β 不可用时，我们通过抽样选取部分评论子集 $\hat{\mathbf{r}}$ ，与专属性 γ 相结合，作为输入替代。这样的设计确保了即使缺乏直接描述，也能通过用户反馈间接获取信息。通过合并商品描述和用户评论，我们构造的输入提示 $f_v(\cdot)$ 提供了完整且有针对性的信息，使LLMs能够生成精确、引人入胜的商品简介。

User Promnt Construction

在生成用户 u 的prompt⁺时，我们利用用户与其互动过的商品的协同信息。首先，我们从用户 \mathcal{I}_u 的互动记录中随机选取子集 $\hat{\mathcal{I}}_u$ 。对每个选中的商品 v ，我们构建一个复合向量 \mathbf{c}_v ，包含元素 α （可能的原始描述）、商品的属性 \mathcal{P}_v 和用户 u 对该商品的评论 r_u^v 。生成用户简介的输入提示 \mathcal{Q}_u 如下：

$\mathcal{Q}_u = \text{"Please generate a user profile for user"}$

$\text{based on their interactions with the following items :"} \cup_{v \in \hat{\mathcal{I}}_u} \mathbf{c}_v$

这个提示要求模型利用用户在 $\hat{\mathcal{I}}_u$ 内商品的互动历史，生成一个精准体现其兴趣和行为特点的个人资料。

$$\mathcal{Q}_u = f_u(\{\mathbf{c}_v | v \in \hat{\mathcal{I}}_u\}).$$

函数 $f_u(\cdot)$ 的角色与 $f_v(\cdot)$ 类似，它将文本内容转化为连贯的描述。每个文本属性 \mathbf{c}_v 源于用户的真实评论，直接展示了他们的个人见解。这种基于用户互动信息的输入提示策略，为我们揭示了用户真实喜好的重要线索。

Density Ratio Modeling for Mutual Information Maximization

在这个阶段 $f(\mathbf{s}_i, \mathbf{e}_i)$ 模型的策略，目的是优化互信息 $I(\mathbf{s}_i, \mathbf{e}_i)$

其中 \mathbf{s}_i 是用户/项目互动的语义表示 \mathbf{e}_i 是相应的简介。鉴于我们已生成了用户和项目的简介 $\mathcal{P}_{u/v}$ ，它们反映了用户与目标的互动偏好，我们的做法是利用这些简介来转化语义信息，具体操作如下：

$$\mathbf{s}_u = \mathcal{T}(\mathcal{P}_u), \quad \mathbf{s}_v = \mathcal{T}(\mathcal{P}_v).$$

1. 对比性建模：这种方法通过拉-推双向对比来有效地对齐不同视角，验证了其在多角度对齐上的有效性。
2. 掩码重建生成性建模：这是一种自监督学习策略，通过从原始数据中恢复被遮盖部分来增强对输入信息的理解，以此来优化CF侧和LLMs的语义表示对齐，减少噪声干扰。

通过这两者，我们旨在确保互动信息在两者间的准确传递，提升推荐系统的性能。

Contrastive Alignment

如图所示，我们定义 $f(\mathbf{s}_i, \mathbf{e}_i)$

为对比对齐（Contrastive Alignment），这是将文本嵌入技术 \mathcal{T} 应用的实际操作，通过对比 \mathbf{s}_i 和 \mathbf{e}_i 来量化它们的相似性，以此来优化CF侧的表示和LLMs的语义表示的对齐，减少噪声影响。

使用余弦相似度⁺函数 $\text{sim}(\cdot)$ 作为相似度量 σ_{\downarrow} 是一个多层感知器⁺，它将 \mathbf{s}_i 映射到与 \mathbf{e}_i 匹配的特征空间⁺。在我们的对比对齐方法中 \mathbf{e}_i 和 \mathbf{s}_i 被视为正样本对。在训练过程中，我们试图通过优化使得正样本对更紧密，即拉近它们的表示，同时在负样本中保持一定的距离。这意味着在每个批次的学习中，我们不仅对正样本进行增强，还要处理并平衡负样本，以确保模型能准确理解和学习两者之间的关联。

Generative Alignment

受到Masked Autoencoder（MAE）这种生成式自监督学习模式的启发，我们创新地将MAE应用于密度比的建模。这样做的目的是在MAE的框架下，通过预测和重构隐含的文本信息，来优化对用户/项目交互信息的表达，进而实现对互动表示和语义表示的有效对齐。

我们借鉴Masked Autoencoder（MAE）的思路，采用 σ_{\uparrow} 作为多层感知模型，对样本的初始嵌入进行语义特征空间映射。

$\mathbf{x} \setminus \mathbf{x}_i$ 代表被遮罩的第 i 个样本。建模过程采用单向重构，仅关注重构被遮罩部分，以提升对语义信息的理解。通过随机遮罩和重构，我们在特征空间中探索了模型的重构能力。通过对比和生成对齐策略，我们确保了语言模型知识与用户偏好的语义一致性。此方法结合了基于身份的协同关系和基于文本的行为语义，分别命名为RLMRec-Con和RLMRec-Gen。在实际数据上的评估中，我们将全面考察这两种模型在不同任务上的表现，揭示各自的优势和局限性。

Model-agnostic Learning

我们主要关注提升[协同过滤⁺](#)（CF）和语言模型（LLM）的表示质量，这两部分都能通过现有模型的用户/项目学习进行优化。因此，我们的方法适用于广泛的推荐系统，通过一个通用的优化目标 $\mathcal{L}_{\mathcal{R}}$ 。总优化目标 \mathcal{L} 包括两部分：

$$\mathcal{L} = \mathcal{L}_{\mathcal{R}} + \lambda \cdot \mathcal{L}_{\text{align}}$$

其中 λ 是调节参数，决定对协同过滤优化和对齐两目标的相对重视程度。 $\mathcal{L}_{\text{align}}$ 是对比对齐（RLMRec-Con）和生成对齐（RLMRec-Gen）的目标函数，通过集成这两方面的改进，我们期望整体上提高推荐系统的效能。

通过最小化总优化函数 \mathcal{L} ，我们追求的是实现提升互信息的目标。这通过联合优化 $\mathcal{L}_{\mathcal{R}}$ （源自推荐系统的损失）和 $\mathcal{L}_{\text{align}}$ （对比对齐及生成对齐的损失）来实现。这样做旨在使模型更深入地理解用户与项目交互的复杂性，从而提高推荐的精度和相关性。

Experimental Settings

Datasets

我们在三个公开数据源上对：

Amazon-book，包含用户对书籍的评分和评论；

Yelp，记录了各类商业实体的详细[文本分类⁺](#)；

Steam，是关于电子游戏用户反馈的数据集。

尽管Amazon-book和Yelp我们都遵循类似的数据预处理，如去除评分低于3的记录，但Steam因不包含评分，我们并未过滤。然后，我们进行k-core过滤，以3:1:1的比例划分训练集、验证集和[测试集⁺](#)。详情可参照附录中的表格，了解各数据集的具体统计信息。

Evaluation Protocols and Metrics

为了确保全面评估和消除偏见，我们在所有实验中采用全排名协议，精确地评估RLMRec。我们利用Recall@N和NDCG@N两个标准指标，它们分别衡量模型在Top-N推荐中的准确性和排序质量，特别关注长尾情况下的表现。Recall@N关注找到目标项的精确率，NDCG则考虑了推荐列表的整体排序效果。这些指标为我们提供了定量的依据，来评估。

Base Models

我们将RLMRec，来检验其效果。SSLRec以其出色的自学能力和对用户行为表示的深入理解著称。这样做的目的是不仅验证，还能对比它与SSLRec等竞品在推荐精度、相关性以及用户体验方面的性能。这种方法让我们得以全面考察，并为它在行业标准中的位置提供实证支持。

- **GCCF**研究了GNN中非线性处理的重构，从而简化了基于图的推荐系统构建过程。
- **LightGCN**通过去除GNN中不必要的神经模块，创建了一个结构简洁的推荐系统。
- **SGL**通过引入节点/边的随机丢弃（dropout）策略，作为一种数据增强手段，生成不同视角来进行对比性学习。
- **SimGCL**通过引入无增广视角生成策略，提升了推荐系统的性能。
- **DCCF**通过分化对比学习来捕获推荐中意图之间的关联关系。
- **AutoCF**是一个自监督的掩码[自编码器⁺](#)，专为推荐系统设计，自动执行数据增强过程。

Implementation Details

对于所有基础模型，我们设定嵌入维度为32。通过网格搜索，我们确定每个模型的超参数。我们通过OpenAI的ChatGPT（如gpt-3.5-turbo）生成用户和商品的特征向量。使用text-embedding-ada-002来获得语义表示 \mathbf{s} 。在训练时，批量大小为4096，学习率设为1e-3，采用Adam优化器。我们采用基于验证集性能的早停策略来指导模型训练。

Performance Comparison (RQ1)

1. RLMRec，展现其普遍适用性，无论在何种基础模型上都能提升准确性和相关性。
2. 虽然增益可能因初始条件略有不同，但总体上，RLMRec，保证了推荐的稳定性。
3. 在应对噪声数据时，RLMRec，推荐准确度不受噪声影响。
4. 尽管增加了复杂性，但RLMRec，证实其高效性。
5. RLMRec，验证了其在不同领域的适应性。这些结果强化了它作为通用推荐系统增强工具的有效性和重要性。
6. 一方面，（语言模型）驱动的精细用户/项目特征建模，增强了从用户交互中捕获的丰富语义细节；另一方面，我们采用跨视图互信息最大化策略，强化了协同过滤部分的关联嵌入与LLM生成的语义表示之间的互动，从而有效去除了推荐特征中的冗余噪声。
7. 显而易见，无论是对比学习还是生成模型+都能显著提升性能。然而，实验结果显示，当与GCCF和SimGCL这类基线相比，对比学习方法在某些情况下表现更出色。而在AutoCF（如AutoCF中需要掩码重建的任务）上，RLMRec-Gen则展现出更大的改进。我们推测，这种结合掩码操作的生成方法，由于其正则化作用，可能在特定任务上能获得更好的效果。

Table 1: Recommendation performance Imprvement of all backbone methods on different datasets in terms of Recall and NDCG. The superscript * indicates the Imprvement is statistically significant where the p-value is less than 0.05.

Data		Amazon-book						Yelp						Steam					
Backbone	Variants	R@5	R@10	R@20	N@5	N@10	N@20	R@5	R@10	R@20	N@5	N@10	N@20	R@5	R@10	R@20	N@5	N@10	N@20
Semantic Embeddings Only	Base	0.0081	0.0125	0.0199	0.0072	0.0088	0.0112	0.0013	0.0022	0.0047	0.0014	0.0018	0.0026	0.0033	0.0062	0.0120	0.0031	0.0043	0.0064
	RLMRec-Con	0.0537	0.0872	0.1343	0.0537	0.0653	0.0807	0.0390	0.0652	0.1084	0.0451	0.0534	0.0680	0.0500	0.0826	0.1313	0.0556	0.0665	0.0830
	RLMRec-Gen	0.0551*	0.0891*	0.1372*	0.0559*	0.0675*	0.0832*	0.0393	0.0654	0.1074	0.0454	0.0535	0.0678	0.0532*	0.0874*	0.1383*	0.0588*	0.0702*	0.0875*
	Best Imprv.	↑4.28%	↑3.10%	↑3.87%	↑4.66%	↑3.98%	↑4.34%	↑4.87%	↑5.06%	↑5.54%	↑5.10%	↑5.24%	↑5.74%	↑7.60%	↑6.90%	↑6.47%	↑7.37%	↑7.22%	↑6.99%
LightGCN	Base	0.0570	0.0915	0.1411	0.0574	0.0694	0.0856	0.0421	0.0706	0.1157	0.0491	0.0580	0.0733	0.0518	0.0852	0.1348	0.0575	0.0687	0.0855
	RLMRec-Con	0.0608*	0.0969*	0.1483*	0.0606*	0.0734*	0.0903*	0.0445*	0.0754*	0.1230*	0.0518*	0.0614*	0.0776*	0.0548*	0.0895*	0.1421*	0.0608*	0.0724*	0.0902*
	RLMRec-Gen	0.0596*	0.0948*	0.1446*	0.0605*	0.0724*	0.0887*	0.0435*	0.0734*	0.1209*	0.0505	0.0600*	0.0761*	0.0550*	0.0907*	0.1433*	0.0607*	0.0729*	0.0907*
	Best Imprv.	↑6.67%	↑5.90%	↑5.10%	↑5.37%	↑5.76%	↑5.49%	↑5.70%	↑6.80%	↑6.31%	↑5.50%	↑5.86%	↑5.87%	↑6.18%	↑6.46%	↑6.31%	↑5.74%	↑6.11%	↑6.08%
SGL	Base	0.0687	0.0994	0.1473	0.0632	0.0756	0.0913	0.0432	0.0722	0.1197	0.0501	0.0592	0.0753	0.0565	0.0919	0.1444	0.0618	0.0738	0.0917
	RLMRec-Con	0.0655*	0.1017*	0.1528*	0.0632*	0.0778*	0.0945*	0.0452*	0.0763*	0.1248*	0.0530*	0.0626*	0.0790*	0.0589*	0.0956*	0.1489*	0.0645*	0.0748*	0.0950*
	RLMRec-Gen	0.0644	0.1015	0.1537*	0.0648*	0.0777*	0.0947*	0.0467*	0.0771*	0.1263*	0.0537*	0.0631*	0.0798*	0.0574*	0.0940*	0.1476*	0.0629*	0.0752*	0.0934*
	Best Imprv.	↑2.83%	↑2.31%	↑4.34%	↑3.16%	↑2.91%	↑3.72%	↑8.10%	↑6.79%	↑5.51%	↑7.19%	↑6.59%	↑5.98%	↑5.20%	↑4.03%	↑3.12%	↑4.37%	↑4.07%	↑3.60%
SimGCL	Base	0.0618	0.0992	0.1512	0.0619	0.0749	0.0919	0.0467	0.0772	0.1254	0.0546	0.0638	0.0801	0.0564	0.0918	0.1436	0.0618	0.0738	0.0915
	RLMRec-Con	0.0633*	0.1011*	0.1552*	0.0633*	0.0765*	0.0942*	0.0470	0.0784*	0.1292*	0.0546	0.0642	0.0814*	0.0582*	0.0945*	0.1482*	0.0638*	0.0760*	0.0942*
	RLMRec-Gen	0.0617	0.0991	0.1524*	0.0622	0.0752	0.0925*	0.0464	0.0767	0.1267	0.0541	0.0634	0.0803	0.0572	0.0929	0.1456*	0.0627*	0.0747*	0.0926*
	Best Imprv.	↑2.43%	↑1.92%	↑2.65%	↑2.26%	↑2.14%	↑2.50%	↑0.64%	↑1.55%	↑3.03%	-	↑0.63%	↑1.62%	↑3.19%	↑2.94%	↑1.53%	↑3.24%	↑2.98%	↑2.95%
DCCF	Base	0.0662	0.1019	0.1517	0.0658	0.0780	0.0943	0.0468	0.0778	0.1249	0.0543	0.0640	0.0800	0.0561	0.0915	0.1437	0.0618	0.0736	0.0914
	RLMRec-Con	0.0665	0.1040*	0.1563*	0.0668	0.0798*	0.0968*	0.0486*	0.0813*	0.1321*	0.0561*	0.0663*	0.0836*	0.0572*	0.0929*	0.1459*	0.0627*	0.0747*	0.0927*
	RLMRec-Gen	0.0666	0.1046*	0.1559*	0.0670*	0.0801*	0.0969*	0.0475	0.0785	0.1281*	0.0549	0.0646	0.0815	0.0570*	0.0918	0.1430	0.0625	0.0741	0.0915
	Best Imprv.	↑0.60%	↑2.65%	↑3.03%	↑1.82%	↑2.69%	↑2.76%	↑3.85%	↑4.50%	↑5.76%	↑3.31%	↑3.59%	↑4.50%	↑2.14%	↑1.53%	↑1.53%	↑1.46%	↑1.49%	↑1.42%
AutoCF	Base	0.0689	0.1055	0.1536	0.0705	0.0828	0.0984	0.0469	0.0789	0.1280	0.0547	0.0647	0.0812	0.0573	0.0933	0.1358	0.0672	0.0684	0.0855
	RLMRec-Con	0.0695	0.1083*	0.1586*	0.0704	0.0837	0.1001*	0.0488*	0.0814*	0.1319*	0.0562*	0.0663*	0.0835*	0.0583*	0.0945*	0.1405*	0.0683*	0.0717*	0.0862*
	RLMRec-Gen	0.0693	0.1069*	0.1581*	0.0701	0.0830	0.0996	0.0493*	0.0828*	0.1330*	0.0572*	0.0677*	0.0848*	0.0579*	0.0888*	0.1410*	0.0599*	0.0710*	0.0864*
	Best Imprv.	↑0.87%	↑2.65%	↑3.26%	↑0.14%	↑1.87%	↑1.73%	↑5.12%	↑4.94%	↑3.91%	↑4.57%	↑4.64%	↑4.31%	↑4.05%	↑4.10%	↑3.83%	↑3.67%	↑3.80%	↑3.63%

相较于LLM增强方法，RLMRec：

1. 对比中，尽管LLM如KAR通过语言模型处理用户/项目描述，但我们的方法更注重语义表示与行为表示的对齐，减少了潜在的无关噪声影响。

1. 我们使用LightGCN和SGL作为基准，确保了对比的公正性，但KAR可能因为直接依赖于语言模型，而忽视了对用户行为特性的直接建模。
2. RLMRec，可能更准确地捕捉到用户行为的内在规律，因此在实际应用中表现出更优性能。

Ablation Study (RQ2)

适应性和鲁棒性+：RLMRec，其强大的语义表示使其能更好地适应用户行为的变化和噪声干扰，展现出更强的稳定性和抗干扰能力。这源于LLMs驱动的精准特征建模和跨视图信息融合，使得模型能抵御潜在的不相关信息影响，从而保证推荐的准确性。

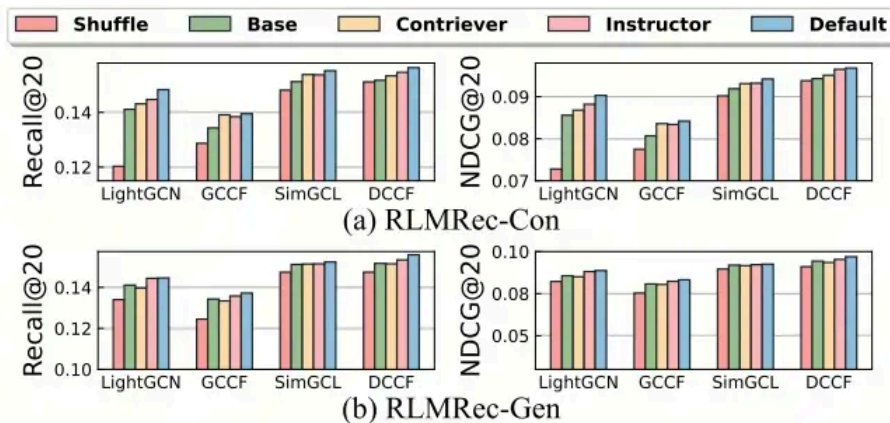


Figure 4: Ablation study on variant text embedding models conducted on the Amazon-book dataset. Shuffling involves reordering user/item embeddings.

在随机扰乱语义表示以消除协同和语义关联时，我们观察到，不论是RLMRec-Con还是RLMRec-Gen，对于基准模型，性能都显著下降。这说明混乱的语义表示引入了由于语义信息与协同信息脱节导致的噪音。这强有力地证实了LLM的语义知识与用户间实际关系精确匹配对于提升推荐效能的重要性。

当我们使用Contriever和Instructor等变体的文本嵌入模型时，即使不改变，默认的text-embedding-ada-002设置，RLMRec，这表明它能有效地利用文本嵌入模型，将文本语义转换为偏好表示，增强推荐性能。这可能是因为这些模型能更精准地捕获语义信息，从而导致性能提升更为显著。

Case Study

我们研究了如何利用LLM整合以捕捉那些通过直接消息难以捕捉的全局用户关系。具体到图示例子，比如用户 u_{1998} 和 u_{227} ，他们之间超过3跳的距离。为了衡量模型在这方面的能力，我们比较了在LightGCN和RLMRec-Con基础上用户表示的相似性。我们采用了相同的基线，同时引入了两个评估指标。

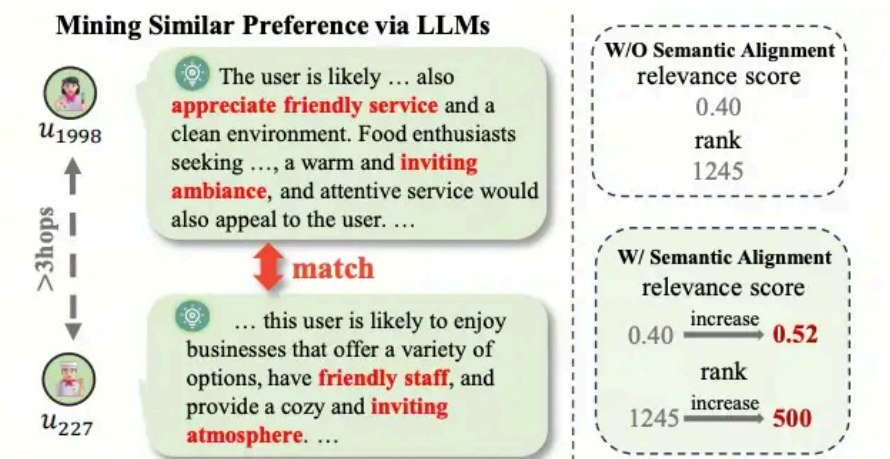


Figure 6: Case study on capturing global user dependencies.

原文《Representation Learning with Large Language Models for Recommendation》

发布于 2024-04-17 10:49 · IP 属地北京

LLM 推荐系统 百度



理性发言，友善互动