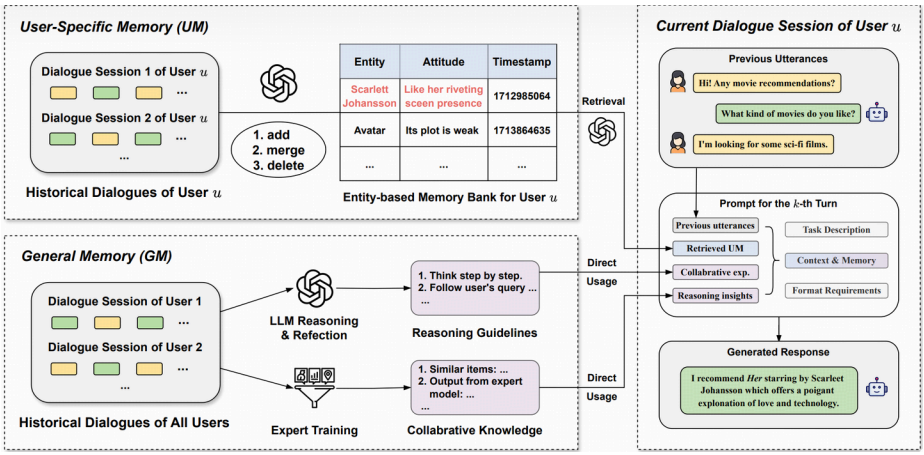
知乎



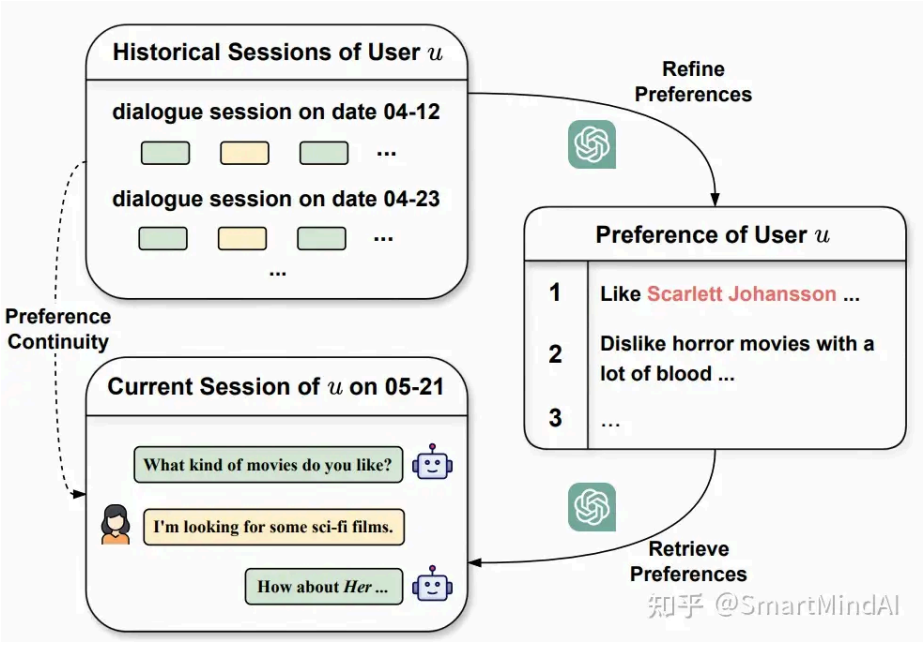
华为2024：MemoCRS——推荐系统中利用LLM捕捉并满足用户的连续性偏好

SmartMindAI
专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我
已关注

Introduction

对话推荐系统（CRS）与用户进行多轮对话交流，旨在获取用户偏好并提供个性化推荐。与仅依赖用户与内容之间的交互的传统推荐系统不同，CRS能够理解自然语言指令，收集实时用户反馈，并通过持续对话推断用户偏好。因此，期望能够提供与人类对话类似的、精确匹配用户需求的推荐内容。为了实现这一目标，CRS通常包含两个核心组件：一个推荐器，用于提供与用户偏好相匹配的推荐；一个生成器，用于产生自然语言响应。

然而，大多数CRS的主要关注当前对话会话中的对话理解和偏好挖掘上，忽略了在历史对话会话中反映的用户偏好。在序列CRS中，用户可能在不同会话中展现出不同的偏好，这些偏好在不同会话中并存且相互关联。整合这些历史偏好有助于我们更好地理解当前对话会话，并揭示一些微妙和隐含的兴趣。例如，在图中，如果用户的历史会话显示对女演员斯嘉丽·约翰逊的偏好，现在用户询问科幻电影，那么由斯嘉丽·约翰逊主演的科幻电影，如"她"，很可能会很好地满足用户的需求和偏好。



近期，诸如ChatGPT这样的大型语言模型（LLMs）在理解与生成自然语言方面展现出了卓越的能力。尽管已有几项关于将LLMs应用于客户关系系统（CRS）的研究探索，但这些工作并未涉及通过历史对话来建模用户偏好连续性的内容。它们通常仅关注于在当前会话中进行推荐、评估CRS作

个性和偏好。因此，我们首次提出将增强记忆的LLMs引入序列式CRS，使得通过可更新的文本记忆机制来建模用户偏好连续性成为可能。

虽然大型语言模型（LLM）是构建插拔式、可解释、透明和可扩展的记忆机制的热门选择，但对于基于LLM的连续性推荐系统（CRS），它通常面临着以下两个关键挑战。

首先，用户的历史对话会通常包含冗余、无关和噪音信息，简单地构建一个包含所有历史对话的记忆库并不理想。通过LLM的提炼，历史对话可以被简化为精炼的偏好知识，以去除冗余。并非所有偏好都与用户当前寻求科幻电影的对话相关。例如，对恐怖电影的偏好可能会引入噪音，因为它与用户的当前需求无关。因此，提炼偏好并检索相关偏好是至关重要的。

其次，推荐不仅依赖于用户的个人偏好，还依赖于用户之间共享的普遍知识，例如协作知识。这种知识需要对数据分布有全面的理解，这是LLM经常难以应对的方面。此外，并非所有用户都有足够的历史对话来建立个性化的记忆，导致冷启动用户的问题，这些用户的记忆有限。因此，保留用户之间的通用知识也是至关重要的。

为了解决上述问题，我们提出了一种以大型语言模型增强的连续性CRS框架（MemoCRS），以捕捉用户偏好连续性，从而为连续性CRS系统提供支持。具体来说，我们设计了两种类型的文本记忆：（1）用户特定记忆和（2）通用记忆。用户特定记忆针对每个用户，为个人和个性化的偏好提供定制支持。我们通过实体记忆库实现这一点，实体库包含历史对话中提到的内容名称、属性以及相关的用户态度和时间戳。这个结构化记忆库支持"添加"、"合并"、"检索"和"删除"等操作。当发生新的对话时，LLM从这个库中检索相关记忆来协助推荐，从而减少冗余和噪音。通用记忆包含共享和普遍知识，跨越了单个对话的界限。在我们的框架中，我们主要关注两个核心方面：（1）协作知识，包含不同用户之间的共享偏好模式，以及（2）推理指南，指导LLM的推理过程。前者由外部专家模型提供，后者则是由LLM自我反思总结的。LLM可以利用这些外部和自我总结的知识为用户，特别是冷启动用户，提供合适的推荐。最后，通过整合这些记忆和当前对话，LLM被赋予了为每位用户提供更精确和定制化推荐的能力。

Preliminaries

对话推荐系统（CRS）旨在通过多轮自然语言对话来引导用户表达偏好，并提供与用户偏好匹配的精确内容推荐。因此，CRS通常由推荐器和生成器组成，推荐器生成与用户偏好匹配的推荐，生成器则基于推荐生成自然语言响应。为了实现这一目标，推荐器和生成器都需要理解对话内容并挖掘用户的偏好。

在实际应用中，用户可能与CRS进行多轮对话，其历史对话和偏好具有序列性和连续性。我们将此称为序列对话推荐系统，形式如下：设$\mathcal{U}$、$\mathcal{I}$、$\mathcal{V}$分别表示用户集、内容集和词汇集。对于用户$u \in \mathcal{U}$，他/她与系统进行T次对话会话，我们将所有他的/她的对话会话按照时间顺序组织起来，并将其表示为$\{C_t\}_{t=1}^{T}$。每个对话C_t由n个陈述组成，表示为$C_t = \{s_k\}_{k=1}^{n}$，其中s_k代表第k轮的陈述，每个陈述s_k由词汇集$\mathcal{V}$中的m个单词组成，即$s_k = \{w_j\}_{j=1}^{m}$。

在每个陈述s_k中，用户或推荐器可能会提及一些内容，表示为 $\mathcal{I}_k \in \mathcal{I}$

我们将当前对话会话的最后一轮（推荐器应该发言的轮次）视为"当前对话会话"，我们的目标是在此提供推荐。那么，表示为"历史对话会话"的所有会话在当前对话会话之前，即用户u的会话集合$H_u = \{C_t\}_{t=1}^{T-1}$。

值得注意的是，与之前的文献相比，我们对"历史对话会话"的定义与大多数CRS文献中常用的"当前对话会话的历史"有所不同。当前对话会话的历史通常指的是同一会话中当前陈述之前的陈述，关注于轮次级别。而历史对话会话则指的是在当前会话之前，不同时间发生的完整对话。 基于上述定义，序列对话推荐任务可以被定义如下。在第k轮（推荐器应发言的轮次），给定用户u的历史对话会话$H_u = \{C_t\}_{t=1}^{T-1}$和当前对话会话在第$k$轮之前的陈述序列$\{s_j\}_{j=1}^{k-1}$，CRS需要从整个内容候选集$\mathcal{I}$中选择一个候选集$\hat{\mathcal{I}}_k$，使得$\hat{\mathcal{I}}_k$尽可能符合用户的当前需求和隐含偏好，并为$\hat{\mathcal{I}}_k$中的内容生成符合需求的响应。

Methodology

知乎

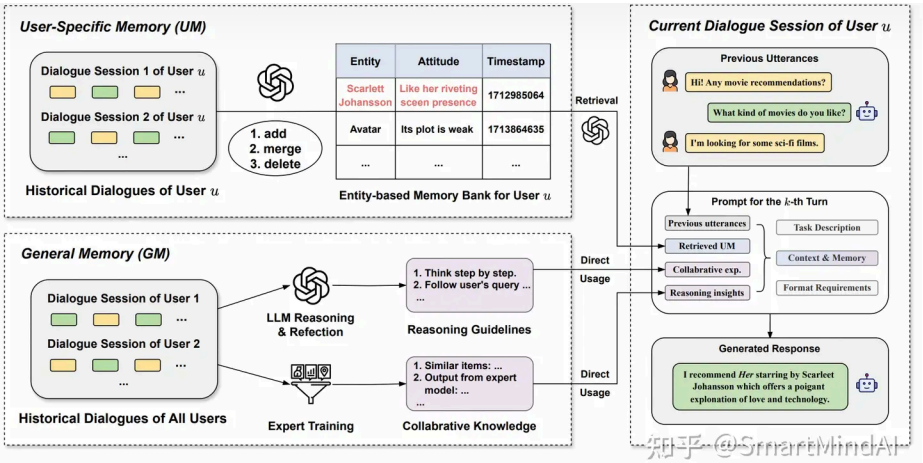


Figure 2: The overall framework of MemoCRS.

MemoCRS的这一框架，如图所示，是模型无关的，由两种类型的记忆组成：（1）用户特定的记忆，用于存储用户的个性化偏好；（2）通用记忆，用于存储用户之间的共享和普遍知识。

用户特定记忆（UM）是为每个用户设计的独特*实体基础记忆库*，用于存储用户的个体化偏好。它是一个动态的记忆库，存储用户历史对话中提到的实体（例如，物品名称和属性），以及与这些实体相关的用户态度和时间戳。它还具有可扩展性和可更新性，支持多种记忆操作，如*添加*、*合并*、*检索*和*删除*。

通用记忆（GM）保留了多样用户之间无法仅从个体对话中得出的共享和普遍知识。它主要由与推荐相关的*协作知识*和LLM推理的*推理准则*组成。这两种知识保持紧凑，可以直接使用，无需从记忆中检索。这种知识的普遍性使LLM能够从对话历史记录有限的冷启动用户中推断出偏好，从而增强推荐。

在处理对话推荐时，通用记忆以及从用户特定记忆中检索的相关实体和态度会被整合到提示中，同时考虑当前对话的上下文。这种整合使得LLM能够生成与用户当前需求和隐含偏好相一致的推荐。

User-Specific Memory (UM)

用户在不同对话中的偏好各不相同，但这些偏好是连续的且同时存在的，这是因为用户行为的一致性。这种"偏好连续性"描述了用户在时间维度上的偏好行为的连贯性和倾向性。理解用户的历史会话偏好有助于洞察他们当前对话中的隐性需求，从而提供更加符合他们需求的推荐。例如，如果用户的历史对话显示偏好为演员"斯嘉丽·约翰逊"，现在用户询问一些科幻电影，那么一部由"斯嘉丽·约翰逊"主演的科幻电影，如《她》，可能是一个很好的选择。

最近受到关于记忆的研究的启发，我们提出利用外部记忆库来记录每个用户的历史偏好，从而帮助LLMs提供更加个性化的推荐。然而，仅仅保存所有历史对话或内容作为记忆并不实际。我们指出，由于用户偏好可能具有多面性，用户在当前对话中往往有特定的需求，因此并非所有历史会话或内容都对当前对话有帮助。这种方法可能会引入不必要的冗余和噪音。此外，当用户有大量历史对话时，包括所有这些对话可能超过LLMs的上下文窗口大小。

一些研究人员也发现，即使上下文长度远未达到LLMs的上下文限制，LLMs在从长用户行为序列的文本上下文中提取有用信息进行推荐时往往失败。因此，我们需要压缩和精炼用户的历史对话会话，只提取与用户当前对话需求相关的记忆。 因此，我们设计了一个基于实体的记忆库，用于存储关键信息，包括实体、用户对实体的态度以及时间戳。LLMs从用户的历史会话中提取实体，如电影标题、演员、导演和电影场景的类型，以及用户对这些实体的态度。此外，这个记忆库是动态的且可扩展的，支持添加、合并、检索和删除等操作。

Memory 记忆库

在用户专属的记忆中，每个用户u需要一个单独的记忆库 $\mathcal{M}_u$，专门用于存储其偏好。我们通过回顾历史对话会话提取与用户偏好紧密相关的关键数据：用户提到的实体、用户对这些实体的微妙态度以及时间戳。这里，我们将记忆库 $\mathcal{M}_u$表示为字典，其中实体作为字典的键，其他信息作为其对应的值。

由于用户提到的实体数量有限，通过这种方式提取的知识比整个对话更为紧凑且与推荐更为相关。这种方法不仅通过去除冗余来提高存储效率，还便于更新和检索，因为我们可以根据实体找到相关的资源。此外，这种设计汲取了人类记忆的复杂性，通常涉及选择性的压缩和总结，而不是保留每一个细节。正如人类回忆时，特定的关键字会作为重要的提示，帮助回忆特定的故事、想法和感受。

实体在记忆库中充当关键，包括用户在历史会话中提到的物品、物品的属性以及物品的特征。在推荐领域中，用户兴趣主要体现在对物品的偏好及其属性上，这也是个性化记忆需要保留的主要内容，并作为引导我们针对特定用户偏好进行检索的关键线索。 态度指的是用户对特定实体的特定观点和立场。仅仅在记忆库中保留实体是不够的，因为用户对特定实体的偏好可能非常复杂，可能是喜欢或不喜欢，或者涉及微妙的条件评估，例如用户在圣诞节时喜欢观看《真爱至上》。例如，在恐怖电影方面，用户可能不喜欢过于血腥的电影，但喜欢心理刺激的电影。此外，用户偏好是动态的，会随时间变化。因此，除了实体，我们还需要捕捉用户对它的特定且最新的态度。 每个条目的时间戳是需要记录的最后一项，表示每个条目的最后一次操作（包括添加、合并和检索）的时间。这种方法旨在方便后续的删除操作。在存储容量成为问题的情况下，我们根据这些时间戳优先删除，确保高效的内存管理。

Memory Update

随着用户进行更多对话会话，他们的偏好可能会持续演变和更替。因此，我们必须随着新的对话会话的出现，不断更新用户u的记忆库$\mathcal{M}_u$。除了删除操作外，添加和合并操作都是通过语言模型（LLM）实现的。这些提示，包括后续由LLM使用的提示，都包含三个部分：任务描述、上下文和格式要求。任务描述提供了整个任务指令的全面概述。上下文包括需要被纳入提示的输入，例如记忆和当前对话的前言话语。格式要求指定了对输出的某些期望，例如JSON或列表格式，这有助于使用后续的文本解析器提取结果。这样的提示设计灵活多变，适用于各种任务。

添加操作是记忆更新的主要组成部分，涉及从对话中提取实体和态度，并将其纳入记忆库$\mathcal{M}_u$。虽然传统的实体抽取模型能够高效地处理实体抽取任务，但我们的工作进一步提取与这些实体相对应的用户态度，因此我们采用了语言模型（LLM）。对于用户u，每当他们完成一次对话会话C_t，我们都会将对话整合到一个添加提示P_{add}中，并通过LLM进行处理，如下所示：

- 首先，根据对话内容，我们使用任务描述来明确添加操作的目标和规则。
- 然后，我们从对话中提取实体和态度，这些信息构成了上下文的一部分。
- 最后，我们将这些实体和态度按照格式要求整合到提示P_{add}中，通过LLM进行处理，以更新记忆库$\mathcal{M}_u$。

这样的提示设计具有灵活性，适用于各种任务。

$$\{(e_i, a_i)\}_{i=1}^{L} = f_{LLM}(C_t, P_{add})$$

这里的等长的L个实体-态度对集合$\{(e_i, a_i)\}_{i=1}^{L}$表示长度为L的实体-态度对的集合，其中e_i是第i个实体a_i是第i个实体对应的属性。具体地P_{add}是用于执行加法操作的提示模板，包含任务说明，如"给定对话，总结多个实体及其用户对这些实体的态度。实体包括电影标题及其关联属性特征。"以及格式需求，如"输出格式应为JSON格式，实体作为键，态度作为值。"同时，我们为这次生成记录了时间戳ts_{gen}，以方便后续的删除操作。随后，我们将每个实体e_i作为键，其对应的属性a_i和时间戳作为值，将添加到$\mathcal{M}_u$中。对于每个实体与态度的配对(e_i, a_i)，我们有

$$\mathcal{M}_u[e_i] \leftarrow \text{Write}(a_i, ts_{gen})$$

其中执行写操作到内存记忆库 $\mathcal{M}_u$。

当执行合并操作时，意味着正在添加的实体e_i已经存在于内存记忆库$\mathcal{M}_u$中，即提取的实体已经存在于内存记忆库中。在这种情况下，我们需要将新生成的态度a_i与现有的态度$\hat{a_i}$合并。LLMs在等式中也执行了这个合并操作，将a_i和$\hat{a_i}$封装到提示模板P_{merge}中，长短期记忆模型在其中实现了这一功能。

$$a_i^* = f_{LLM}(a_i, \hat{a_i}, P_{merge})$$

其中深度学习模型生成的合并a_i^*，提示模板P_{merge}包含任务描述，如"根据用户现有的和新的态度，合并这两个态度，在存在冲突时，新的态度被优先考虑。"操作完成后，合并时间戳ts_{merge}被

$$\mathcal{M}_u[e_i] \leftarrow \text{Write}(a_i^*, ts_{merge})$$

此外，合并操作还可以从同一实体扩展到几个语义相似的实体，例如"电话"和"手机"，这进一步提高了存储效率。删除操作遵循规则，无需语言模型的介入。此外，它被视为可选的步骤。在存储空间受限时，我们设置时间阈值D，对整个内存记忆库进行周期性扫描。然后，按照时间戳，删除在至少D时间内没有进行过任何操作（即添加、合并或检索）的对象。

$$\mathcal{M}_u \leftarrow \text{Delete}(\mathcal{M}_u, D)$$

其中，删除表示从集合Mu中删除每个记录，如果其时间戳与当前时间的时间间隔大于D。请注意，这仅仅是删除操作的一种实现方式。

Memory Retrieval

既然我们已经为用户u建立了个性化的记忆库，其中的并非所有记忆都一定对当前对话会话C_T的第k轮对话有益。使用所有记忆会引入噪音，并可能导致超过LLMs的上下文窗口，从而影响性能。因此，从记忆库中检索与当前对话会话相关的记忆变得至关重要。先前的记忆检索工作大多采用向量相似性检索方法，如余弦相似度+，这种方法可以快速处理大量数据，但也可能检索到一些无关的内容。LLMs在判断相关性方面表现更好，但在处理大量候选集时有困难。因此，我们结合了两种方法------首先，采用向量相似性获取候选实体列表$\hat{\mathcal{E}}_u$，然后利用LLMs进一步筛选出相关的实体。

具体来说，我们首先使用向量相似性检索方法，例如余弦相似度，作为初步筛选步骤，以获取候选实体列表$\hat{\mathcal{E}}_u$。当记忆库中的实体数量相对有限时，我们可以省略这一步，直接将记忆库$\mathcal{M}_u$中的所有实体作为$\hat{\mathcal{E}}_u$。接下来，我们将$\hat{\mathcal{E}}_u$和当前会话前$k-1$轮的对话以及$\hat{\mathcal{E}}_u$整合到提示模板$P_{retrieve}$中，如等式所示，允许LLMs选择相关实体。

$$\mathcal{E}_u = f_{LLM}(\hat{\mathcal{E}}_u, \{s_j\}_{j=1}^{k-1}, P_{retrieve})$$

在当前对话相关的检索实体列表中，$\mathcal{E}_u$扮演着关键角色。$P_{retrieve}$包含了任务描述部分，例如，要求从实体列表中选择最相关的Q个实体，并按照相关性进行排序，输出结果是列表形式。Q是超参数值。借助$\mathcal{E}_u$，我们能从$\mathcal{M}_u$中获取对应的用户态度列表$\mathcal{A}_u$，如同公式所示。这些精炼和相关的信息将有助于后续部分中的LLMs提供更优质的推荐。

$$\mathcal{A}_u = \text{Read}(\mathcal{M}_u, \mathcal{E}_u)$$

在用户专属的记忆模块中，调用LLMs的频率相对较低。这主要是因为记忆更新操作的数量较少。通常，LLMs只在每次对话会话结束时进行一次添加操作调用，而合并操作则在实体重叠非常轻微的情况下才会偶尔被需要。更不用说删除操作根本不会涉及LLMs。此外，记忆检索操作仅在推荐时被触发，每次检索时，LLMs会在预筛选的候选实体上被调用一次。

General Memory (GM)

尽管我们已经从用户的历史对话中推导出了用户特定的记忆，但这对于对话推荐来说是不够的。一方面，先前的对话代理或助手通常只考虑每个用户独有的用户特定记忆，因为他们不涉及对话推荐固有的推荐任务。这项任务不仅依赖于用户的个人偏好，还依赖于基于相似用户偏好的协作知识，即通过推断来理解用户偏好。这些见解无法从单个用户的历史对话中得出；相反，模型需要全面理解整体数据分布。

另一方面，并非所有用户都有足够的历史对话会话来形成足够的用户特定记忆。这导致了冷启动用户的问题，他们拥有有限的记忆。提高对这些冷启动用户的表现也是一个重要考虑，需要关注。

为了达到这个目标，除了用户特定的记忆之外，我们还需要考虑一些用户之间的共享知识，称为通用记忆。这里主要关注两个方面：协作知识和基于LLM的推理规则+。训练LLM获取嵌入的协作知识需要大量成本。因此，我们整合了一个外部的专业专家模型，专门用于提取协作信号，为LLM提供低成本的协作知识。LLM推理过程中获得的智慧和经验也构成了用户共享知识的重要部分。因此，我们维护了一个存储库来容纳LLM在推理过程中获得的推理知识和经验。这两类知识都是简洁明了，易于使用，无需检索。它们的融合效果可以增强推荐的效率，特别是对冷启动推荐用户非常有利。

推荐任务中的关键要素是协作知识，它汇集了从大量用户行为中提取的共享模式的聚合，帮助推荐系统揭示用户群体之间的共同点和趋势，确保推荐的精准性。此外，它通过分析类似用户群的行为和偏好，为新用户提供了定制化的推荐，有效地解决了冷启动用户的问题。协作知识通常需要模型理解整个数据集的分布情况，这通常通过在全数据集上进行训练来实现。鉴于训练大型语言模型（LLMs）的高成本，我们采用了外部的专业模型来提取这种知识，从而赋予LLMs以成本效益高的协作洞察。

具体地，对于当前会话的第k轮对话C_T，专家模型$g(\cdot)$的输入包括前$k-1$轮的对话陈述$\{s_j\}_{j=1}^{k-1}$，以及在这些轮次中提到的内容 $\{\mathcal{I}_j\}_{j=1}^{k-1}$

其中$\mathcal{I}_j$表示第j轮对话中用户或推荐器提到的内容集合。专家模型根据前$k-1$轮对话和提及的内容预测第k轮推荐。冷启动用户的问题通过分析类似用户群的行为和偏好得以解决，赋予新用户定制化的推荐。全数据集的使用确保了模型能够理解整个数据集的分布情况，从而在推荐系统中提供更精准的推荐。

$$\hat{\mathcal{I}}_k = g(\{s_j\}_{j=1}^{k-1}, \{\mathcal{I}_j\}_{j=1}^{k-1})$$

Reasoning Guidelines

考虑到我们利用LLM进行推荐，LLM的推理准则成为了另一种关键的知识，应该在用户之间共享。先前的研究发现，从各种决策任务中提取自然语言经验对后续任务是有益的。因此，我们持续地引导LLM反思当前的推理过程，从成功或失败的例子中提取经验，并利用这些经验来辅助后续的推理任务。具体来说，我们首先提供一个由人工制作的简单推理指南集$\mathcal{R}$作为初始化。这个集合包括基本的推理规则，如"让我们一步一步思考"和"在对话中考虑用户的需求"。随后，我们将LLM的推理轨迹t和推荐结果o整合到动态提示模板$P_{reflect}$中，允许根据LLM的学习和经验，对不断演进的推理指南集$\mathcal{R}$进行迭代更新。

$$\mathcal{R} \leftarrow f_{LLM}(t, o, \mathcal{R}, P_{reflect})$$

Integration with Memory for CRSs

在介绍了用户特定记忆和通用记忆之后，我们将两种知识整合到提示中，以使语言模型在对话过程中生成推荐。在当前会话的第k轮，我们根据前$k-1$轮的对话输出，从用户u的个性化记忆库$\mathcal{M}_u$中检索相关实体和态度列表$\mathcal{E}_u$和$\mathcal{A}_u$。接下来，我们使用公式从专家模型获取协作知识，即预测的推荐列表$\hat{\mathcal{I}}_k$。这些信息，加上之前的对话输出，以及推理指导集$\mathcal{R}$，都被整合在提示P_{rec}中，并被输入到语言模型以获取推荐。

$$\widetilde{\mathcal{I}_k} = f_{LLM}\left(\{s_j\}_{j=1}^{k-1}, \mathcal{E}_u, \mathcal{A}_u, \hat{\mathcal{I}}_k, \mathcal{R}, P_{rec}\right)$$

其中$\widetilde{\mathcal{I}_k}$表示由LLM生成的推荐内容列表。基于用户的对话、推理指导原则和历史记忆，提示模板P_{rec}可以利用专家模型和LLM的能力，任务描述如下："从专家模型推荐的电影列表中选择最适合用户需求的前20部电影。若专家模型推荐的电影少于20部，根据自己的知识进行补充，以确保覆盖所有相关电影。"

Experiment Setups

Dataset

我们在中国数据集TGReDial 和英语数据集ReDial上进行实验。TGReDial是通过半自动的话题引导方式构建的中国对话式推荐会话集合，包含10,000个会话，共129,392次对话，涉及1,482个用户和33,834部电影。ReDial是由通过亚马逊机械土耳其的众包工作者手动构建的英语对话式推荐数据集。它包括了与51,699部电影和504个用户相关的10,006次对话，共182,150次对话。

Effectiveness Comparison (RQ1)

Recommendation Task

知乎

Model	HR			MRR			NDCG			HR			MRR			NDCG		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
ReDial	0.0030	0.0055	0.0102	0.0015	0.0018	0.0021	0.0018	0.0027	0.0038	0.0293	0.0413	0.0882	0.0174	0.0190	0.0223	0.0203	0.0242	0.0361
KBRD	0.0050	0.0112	0.0174	0.0026	0.0035	0.0040	0.0032	0.0053	0.0069	0.0824	0.1395	0.2185	0.0337	0.0418	0.0470	0.0457	0.0646	0.0842
KGSF	0.0112	0.0149	0.0249	0.0039	0.0044	0.0051	0.0057	0.0068	0.0094	0.0908	0.1378	0.2319	0.0385	0.0445	0.0508	0.0514	0.0663	0.0898
TGReDial	0.0075	0.0174	0.0236	0.0055	0.0068	0.0072	0.0059	0.0091	0.0107	0.0874	0.1395	0.2336	0.0433	0.0502	0.0567	0.0543	0.0710	0.0948
UCCR	0.0087	0.0174	0.0286	0.0071	0.0082	0.0090	0.0075	0.0103	0.0131	0.1059	0.1782	0.2672	0.0443	0.0538	0.0596	0.0594	0.0826	0.1047
UniCRS	0.0050	0.0124	0.0236	0.0024	0.0035	0.0042	0.0031	0.0055	0.0083	0.1005	0.1605	0.2480	0.0393	0.0467	0.0528	0.0542	0.0731	0.0952
ZSCRS	0.0025	0.0087	0.0100	0.0007	0.0016	0.0017	0.0012	0.0032	0.0035	0.1261	0.1882	0.2353	0.0511	0.0594	0.0631	0.0655	0.0896	0.1018
MemoCRS	**0.0162***	**0.0261***	**0.0323***	**0.0095***	**0.0108***	**0.0112***	**0.0111***	**0.0143***	**0.0158***	**0.1361***	**0.2151***	**0.2857***	**0.0718***	**0.0821***	**0.0871***	**0.0875***	**0.1128***	**0.1308***

为了评估我们提出的MemoCRS在推荐系统任务的有效性，我们将它与选择的最前沿的CRS基线进行了比较。结果如下表所示，通过这些观察，我们得出以下结论：

- 我们提出的MemoCRS显著超越了基线。例如，在TGReDial数据集上，MemoCRS在HR\@20上的提升为13.04个百分点，在MRR\@20上的提升为23.73个百分点，在NDCG\@20上的提升为20.41个百分点，相对于最强的基线。在ReDial数据集上，这些提升分别为6.93个百分点，38.49个百分点，和24.96个百分点。这表明在CRS中整合增强记忆的LLM和建模用户偏好的连续性对于提高性能具有有效性。
- 基于用户历史的方法通常表现更优。如TGRedial这样的模型，它利用了之前与用户互动过的内容，以及URCC这样的模型，它使用了历史对话会话，优于其他基准模型，获得了更好的结果。MemoCRS通过引入记忆技巧，更精细地调整了用户的行为历史偏好，并证明了在CRS中建模用户偏好连续性和顺序性方法的重要性。
- 使用更大的PLMs往往能带来更好的性能，尽管其性能受到数据集的影响。通过集成LLMs，简单的零射击提示方法（如ZRCRS）在ReDial上的表现尤为出色，这在先前的研究中得到了验证。然而，这一方面同样受到数据集的影响。在TGReDial中，零射击提示的效果明显不佳，这可能是由于LLMs（GPT-4）对中国语言和中国电影的了解有限，加上数据集主要涉及小众电影。

Dialogue Generation Task

除了推荐任务之外，对话生成也是会话推荐系统中至关重要的一个方面。考虑到大型语言模型（LLMs）在语言生成能力上通常远优于在先前的会话推荐系统（CRSs）中使用的较小的预训练语言模型（PLMs），大多数基于LLMs的对话推荐工作主要集中在推荐任务上。一些研究者发现，LLMs在提供可解释的推荐和构建互动用户体验方面表现出色。 在这里，我们通过人工评估，定量比较了在会话推荐系统中使用零次提示生成对话（gpt-4-1106-preview）与传统CRSs的质量。从表 中的结果可以看出，LLMs生成的对话在流畅度和信息量上显著优于传统CRS模型生成的对话。利用LLMs进行CRS可以产生更加自然且准确的响应。

Table 2: Comparison on dialogue generation task.

Model	TGReDial		ReDial	
	Fluency	Informativeness	Fluency	Informativeness
ReDial	0.45	0.40	0.41	0.40
KBRD	1.04	0.99	0.97	1.01
KGSF	1.04	1.08	1.13	1.15
TGReDial	0.80	0.84	0.88	0.87
UCCR	1.10	1.12	1.11	1.17
UniCRS	0.42	0.43	1.18	1.19
LLM	**1.87***	**1.82***	**1.86***	**1.87***

原文《MemoCRS: Memory-enhanced Sequential Conversational Recommender Systems with Large Language Models》

发布于 2024-08-15 11:58 · IP 属地北京

LLM 推荐系统 对话系统

理性发言，友善互动