

# 面向用户偏好建模的个性化对话推荐算法

尚琛展<sup>1</sup>, 赵鑫<sup>2</sup>

<sup>1</sup> 华中科技大学计算机科学与技术学院 武汉 中国 430074

<sup>2</sup> 中国人民大学高瓴人工智能学院 北京 中国 100872

**摘要** 推荐系统的目标是从物品数据库中, 选择出与用户兴趣偏好相匹配的子集, 缓解用户面临的“信息过载”问题。因而近年来推荐系统越来越多地应用到电商、社交等领域, 展现出巨大的商业潜力。传统推荐系统中, 系统对用户的认知往往来源于历史交互记录, 例如点击率或者购买记录, 这是一种隐式用户反馈。对话推荐系统能够通过自然语言与用户进行多轮对话, 逐步深入挖掘其兴趣偏好, 从而向对方提供高质量的推荐结果。相比于传统推荐系统, 对话推荐系统主要有两方面的不同。其一, 对话推荐系统能够利用自然语言与用户进行语义上连贯的多轮对话, 提升了人机交互中的用户体验; 其二, 系统能够询问特定的问题直接获取用户的显式反馈, 从而更深入地理解用户兴趣偏好, 提供更可靠的推荐结果。目前已经有不少工作在不同的问题设定下对该领域进行了探索, 然而尽管如此, 这些工作仍仅局限于关注当前正在进行的对话, 忽视了过去交互记录中蕴涵的丰富信息, 导致对用户偏好建模的不充分。为了解决这个问题, 本文提出了一个面向用户偏好建模的个性化对话推荐算法框架, 通过双线性模型注意力机制与自注意力层次化编码结构进行用户偏好建模, 从而完成对候选物品的排序与推荐。本文设计的模型结构能够在充分利用用户历史对话信息的同时, 权衡历史对话与当前对话两类数据的重要性。丰富的用户相关信息来源使得推荐结果在契合用户个性化偏好的同时, 更具备多样性, 从而缓解“信息茧房”等现象带来的不良影响。基于公开数据集的实验表明了本文方法在个性化对话推荐任务上的有效性。

**关键词** 对话推荐系统; 用户建模; 图神经网络; 注意力机制

**中图法分类号** TP391 **DOI号** 10.19363/J.cnki.cn10-1380/tn.2021.09.05

## Modeling User Preference for Personalized Conversational Recommendation

SHANG Chenzhan<sup>1</sup>, ZHAO Xin<sup>2</sup>

<sup>1</sup> School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

<sup>2</sup> Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China

**Abstract** Recommender Systems aim to select the subset that matches users' preference from the item database, so as to alleviate the problem of “information overload” faced by users. Therefore, in recent years, recommender systems have been increasingly applied to e-commerce, social networking and other fields, showing great commercial potential. In conventional recommender systems, the information of users is often derived from the historical interaction records of items, such as click rates and purchase history, which can be seen as a kind of delayed and implicit feedback from users. Conversational recommender systems (CRS) are able to gradually dig into users' preference through multi-turn dialogues in natural language, so as to recommend high-quality items to them. Compared to conventional recommender systems, CRS exhibit major differences in two aspects. Firstly, CRS are able to interact for multiple semantically coherent rounds with users in natural language, which improves user experience during human-computer interaction. Secondly, it becomes possible for CRS to obtain explicit feedback from users by asking specific questions, which helps system to have a deeper understanding of users' interest and preference, and provide more reliable recommendation results to them. Many works have explored conversational recommendation under different problem settings. However, these works only focus on the ongoing dialogue, ignoring the rich information contained in the past conversations, which leads to insufficiency in user modeling. To address these issues, we propose an end-to-end framework named PCR, which stands for user preference modeling based Personalized Conversational Recommender. PCR is able to model users through the context-attentive layer with bilinear scoring function and the self-attentive hierarchical encoding structure. Our proposed model can make full use of historical dialogues while weighing the importance of the historical dialogues and the current dialogue. More information sources about users make the recommendation results more diversified while conforming to users' preferences, which will alleviate the impact of “information cocoon”. Extensive experiments have demonstrated the effectiveness of our approach on personalized conversational recommendation task.

**通讯作者:** 赵鑫, 博士, 教授, Email: batmanfly@gmail.com

本课题得到国家自然科学基金(No. 61872369, No. 61832017)资助。

收稿日期: 2021-05-06; 修改日期: 2021-08-09; 定稿日期: 2021-08-10

**Key words** conversational recommender system; user modeling; graph neural network; attention mechanism

## 1 引言

近年来,推荐系统在社交<sup>[1]</sup>、电商<sup>[2]</sup>、新闻<sup>[3]</sup>等领域得到了越来越广泛的应用,充分显示了其商业潜力。在传统推荐系统中<sup>[4]</sup>,推荐任务的实现高度依赖于用户与物品的历史交互记录,例如电商网站通过用户的点击记录与购买记录来向用户推送其可能感兴趣的物品。然而,推荐系统却难以从这种隐式反馈中挖掘用户的真实偏好,使得系统性能的提升受到限制。针对该问题,对话推荐系统<sup>[5-7]</sup>能够通过使用自然语言与用户进行多轮对话,逐步深入挖掘其个性化偏好,从而提供高质量的推荐结果。相比于传统推荐系统,用户在与对话推荐系统进行交互时,提供了及时的、显式的反馈,这更精确地反映了用户的真实需求,使推荐系统的性能有更大的提升空间。

在设计上,对话推荐系统由推荐模块与对话模块组成<sup>[8]</sup>。一方面,对话模块能够理解用户意图,向推荐模块提供有关用户偏好的信息,并根据其反馈生成对用户的回复。另一方面,推荐模块根据对话模块提供的信息,在物品数据库中搜寻匹配项,从而更高质量地完成推荐任务。表 1 所示为电影推荐场景下,对话推荐系统与用户交互的示例。系统在与用户交流的过程中关注到其喜欢的电影类型,随后在回复中给用户推荐影片,当用户接受了系统所提供的建议时,整个工作流程结束。在这个过程中可以发现,系统需要同时完成对话和推荐两类任务,因此,做好两个模块的衔接与协同运转对于构建高效的对话推荐系统来说至关重要。

**表 1 电影推荐场景下的对话推荐系统交互示例**  
**Table 1 Example of conversational recommendation in movie recommendation scenario**

角色	对话内容
用户	我想找一部电影和朋友一起看。
系统	你喜欢什么类型的电影呢?
用户	我想找一部科幻片,之前看过《盗梦空间》(2009),让我印象十分深刻。
系统	你可以试一下《星际穿越》(2014),同样是由克里斯托弗·诺兰导演的科幻片。
用户	太好了,谢谢你!

对话推荐系统近年来受到越来越多的关注,针对不同的问题设定,研究者们提出了各种算法来应对不同的需求。例如,ReDial<sup>[5]</sup>引入情感分析模块,根

据用户表达中体现的情感,来更精确地获取其对候选物品的偏好;KBRD<sup>[6]</sup>和 KGSF<sup>[7]</sup>引入外部知识库,来为推荐任务提供充足的信息,缓解了系统所面临的冷启动问题<sup>[9]</sup>;CPR<sup>[10]</sup>将对话推荐任务转化为在图上的路径推理问题,提升了系统性能,并增强了推荐结果的可解释性。

上述的这些工作都在一定程度上提升了对话推荐系统的性能表现,但是它们仍具有局限性。

一方面,现有的工作<sup>[5-7, 10]</sup>都仅对当前的对话进行建模。为了使用户能够在较短时间内获取系统提供的推荐物品,保证良好的用户体验,对话的轮次需要控制在一定的范围内。而这种对交互深入程度的限制,导致系统对用户的认识是片面的,难以深入挖掘用户对候选物品的兴趣偏好。如何在控制对话长度的同时加深系统对用户的理解,这是对话推荐算法需要解决的一个重要问题。

另一方面,从客观上来讲,用户的偏好是广泛的、多方面的,与系统的单次对话仅反映了用户偏好的一方面,在此基础上得到的推荐结果无法体现用户本身丰富的兴趣偏好,具有同质化趋势。这种同质化趋势容易造成“信息茧房”等不良现象,与推荐系统的初衷相违背。

为了解决这两个问题,可以考虑基于用户的历史对话数据进行建模。一方面,在不增加当前对话轮次的前提下,系统可以从用户的历史对话数据中得到更充分的信息,从而获取对用户更深入的理解。另一方面,用户的每次对话可能有不同的主题,丰富的主题展现了用户在不同方面的兴趣偏好,对这些信息加以利用能够缓解当前对话信息的匮乏,提升推荐结果的多样性。

然而,在对用户的历史对话数据进行建模时仍会遇到不小的挑战。首先,并不是所有的对话记录对于当前的推荐任务都有价值。例如在电影推荐场景下,某用户的大多数对话中都向系统寻求属于科幻片的推荐影片,而在当前的对话中他希望寻找喜剧片,如果系统再根据过去的记录给用户推荐科幻片,就会造成严重失误。如何甄别历史对话数据中对当前任务有用的信息,是基于多对话进行用户偏好建模的重要挑战。其次,用户的历史对话数据是丰富的,同时也是冗余的,如果对其不加以区别地利用,势必会导致当前对话中的信息淹没在历史对话数据中,使系统得到与当前任务近乎无关的推荐结果。如何权衡历史对话数据和当前对话数据二者的重要性,

是需要应对的第二个挑战。

为应对这些挑战, 本文提出了 **PCR** (user preference modeling based Personalized Conversational Recommender) 模型。一方面, 使用注意力机制给历史对话中所有和用户发生过交互的物品赋予一定的权重, 权重越大, 表明它与当前任务的契合程度越高, 从而甄别出历史对话数据中最有价值的信息。另一方面, 本文工作采用“目标对话为主, 历史数据为辅”的原则, 基于自注意力机制<sup>[11]</sup>, 设计了层次化自注意力编码器结构, 先对历史对话数据中与用户发生过交互的物品进行融合, 得到与历史相关的用户向量表示, 再将它与当前对话中出现的实体进行融合得到最终的用户向量表示。这样的层次化编码既考虑到用户在多次对话中所展现出的不同方面的兴趣偏好, 又切实关注了他们在当前的对话中所提出的需求, 从而实现对用户偏好的精细建模。

基于公开数据集<sup>[5]</sup>的实验表明, 相对于基线模型, **PCR** 在推荐任务和对话任务上的表现都有所提升, 这表明了本文方法的有效性。

## 2 相关工作

### 2.1 推荐系统

推荐系统的目标是从物品数据库中, 选择出与用户兴趣偏好相匹配的子集, 从而缓解用户面临的“信息过载”问题。推荐系统通常以用户和物品的历史交互记录作为信息来源, 获取匹配的推荐结果, 例如用户的点击记录、观看或购买记录, 甚至是评论等文本信息。

经过 20 余年的发展, 研究者们提出了许多不同的算法来解决推荐问题。早期所采用的方法主要是协同过滤<sup>[12]</sup>, 但是它难以处理稀疏矩阵, 推荐结果的头部效应较为明显。矩阵分解算法<sup>[13]</sup>的提出增强了推荐算法模型的泛化能力, 但它无法对用户、物品及其他上下文特征加以利用。逻辑回归在推荐系统中的应用<sup>[14]</sup>解决了这一问题, 它能够融合多种不同类型的特征, 随后的因子分解机<sup>[15]</sup>可以对不同特征进行进一步的交叉, 增强了模型的表达能力。

深度学习方法兴起以来, 推荐系统领域也受到这一浪潮的深刻影响, 涌现出众多性能优异的新模型<sup>[16-18]</sup>。其后, 为了应对冷启动问题和交互数据稀疏性带来的问题, 研究者们提出了许多技术手段来融入外部信息, 比如异质信息网络<sup>[19]</sup>、知识库<sup>[20-21]</sup>和社交网络<sup>[22]</sup>等, 这也进一步增强了推荐结果的可解释性。

### 2.2 对话推荐系统

传统推荐系统利用历史交互记录来获得关于用户的信息, 在形式上是隐式的, 在时间上是滞后的。对话推荐系统能够通过与用户的多轮对话, 获得显式而及时的反馈, 逐步深入挖掘用户兴趣偏好, 从而提供更高质量的推荐结果。

因此, 对话推荐系统近年来吸引了越来越多研究者的关注。研究者们针对不同的问题设定和形式化方法, 完成了许多有价值的工作。**CRM**<sup>[23]</sup>将深度强化学习方法引入对话推荐系统, 构建了一个深度策略网络来决定系统的下一步动作。有研究者提出了 **SAUR**<sup>[24]</sup> (System Ask, User Respond) 范式来描述基于对话的搜索或推荐问题。对话推荐系统框架 **EAR**<sup>[25]</sup> (Estimation-Action-Reflection) 统一地解决向用户问什么, 何时进行推荐, 如何适应用户的反馈这三个问题。**ReDial**<sup>[5]</sup>是一个端到端的对话推荐算法模型, 该工作同时提出了电影推荐场景下的对话数据集。**KBRD**<sup>[6]</sup>在前者数据集的基础上引入外部知识图谱 **DBpedia**<sup>[26]</sup>, 缓解了冷启动问题, 得到更具可解释性的推荐结果。**KGSF**<sup>[7]</sup>同时引入实体级和词汇级知识图谱, 通过互信息最大化算法来消解实体、词汇两类向量表示之间的语义鸿沟, 进一步提升了对话推荐系统的表现。

然而现有的工作仅对当前对话进行建模, 没有充分考虑同一用户在历史对话中所蕴涵的兴趣偏好信息。基于多对话进行用户建模后, 对话推荐系统具备了更大的性能提升空间。

## 3 问题描述

对话推荐系统的任务是通过多轮对话向用户推荐匹配的物品。对话模块能够分析和理解用户基于自然语言的表达, 并生成相应的回复; 推荐模块能够接收对话模块提供的用户相关信息, 在物品数据库中寻找匹配的推荐结果。多轮对话的过程直到用户接受了推荐结果或者用户离开才会结束。个性化对话推荐任务则更进一步地要求对用户的历史对话数据进行建模, 从而得到更符合用户兴趣偏好的推荐结果。

形式化地说,  $i$  表示候选物品集  $\mathcal{I}$  中的一个物品,  $u$  表示用户集  $\mathcal{U}$  中的一位用户。对于特定用户  $u$ , 当前正在进行的对话  $\mathcal{C}^{(t)}$  称为目标对话, 用户的历史对话按照时间先后顺序排列构成历史对话序列  $\mathcal{H} = \{\mathcal{C}_j\}_{j=1}^{|\mathcal{H}|}$ , 其中  $\mathcal{C}_j$  为该用户与系统发生的第  $j$  次对话。任意对话  $\mathcal{C}$  是由句子构成的序列  $\mathcal{C} = \{s_k\}_{k=1}^{|\mathcal{C}|}$ ,

其中  $s_k$  是对话  $\mathcal{C}$  进行到第  $k$  轮时的句子。

目标对话  $\mathcal{C}^{(l)}$  进行到第  $k$  轮时, 系统需要根据目标对话中已经出现的上下文和历史对话序列, 从物品集  $\mathcal{I}$  中寻找与用户偏好相匹配的子集  $\mathcal{I}_k$ , 并生成回复语句  $s_{k+1}$  作为对用户的反馈。这样的推荐过程直到用户接受了推荐结果或者主动离开为止。

已有的对话推荐系统工作仅利用目标对话中的信息得出推荐结果, 而本文工作充分考虑了历史对话中所提供的丰富信息, 来生成更精确地符合用户真实偏好的推荐结果。

## 4 模型

本文提出了面向用户偏好建模的个性化对话推荐算法框架 **PCR** 来完成个性化对话推荐任务, 该模型首次将用户的历史对话数据作为上下文信息融入到用户建模的过程中。图 1 展示了 **PCR** 模型推荐模块的整体结构, 其中右侧为基于 **KBRD**<sup>[6]</sup> 的基线模型, 左侧为 **PCR** 基于历史对话进行建模的模块, 其得到的与历史相关的用户向量表示能够增强最终用户向量的表达能力, 从而更精确地与物品数据库中的物品进行匹配。

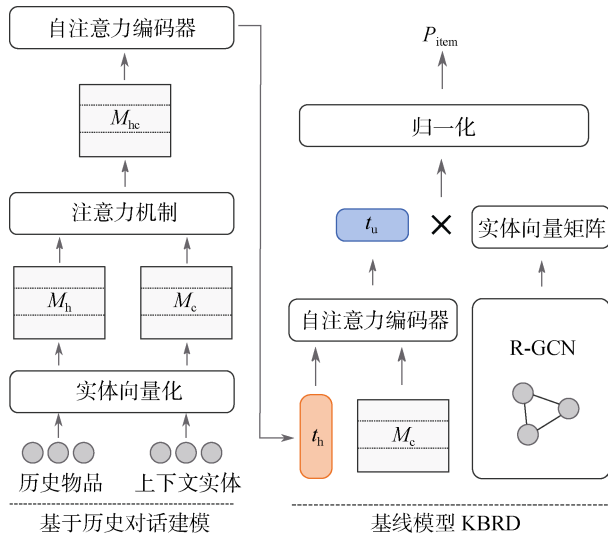


图 1 PCR 推荐模块结构图

Figure 1 Structure of recommendation module in our proposed model PCR

### 4.1 构建实体向量

在基于深度学习的推荐系统中, 通常使用向量的形式来表示一个实体。可以将实体表示为一个独热向量 (One-Hot Vector), 也就是除了标识实体序号维度上的值为 1, 向量其他维度的值都为 0。这种独

热表示方法具有不可忽视的缺点: 一方面其数据具有稀疏性, 使得神经网络训练时的收敛速度变慢, 提高了计算代价; 另一方面, 无法基于这种向量表示来直接比较实体之间的相似程度, 因为任意两个不同实体向量之间的点积都为 0。为了弥补这些缺点, 研究者们通常使用嵌入 (Embedding), 一类低维稠密向量来表示实体, 这能够在降低计算代价的同时提升向量的表达能力。

本文使用 **R-GCN**<sup>[27]</sup>, 一类图卷积神经网络模型来对知识图谱中的结构信息进行编码, 得到节点的低维稠密向量表示。编码时的图结构来源于 **DBpedia**<sup>[26]</sup> 知识库, 整个图结构  $\mathcal{G}$  是三元组  $(e_h, r, e_t)$  的集合, 其中  $e_h, e_t \in \mathcal{E}$ , 分别表示头节点和尾节点对应的实体,  $r \in \mathcal{R}$  表示节点之间的关系, 其中  $\mathcal{E}$  和  $\mathcal{R}$  分别表示知识图谱中的实体集和关系集。根据图结构的信息, 可以训练实体向量所构成的参数矩阵。

形式化地, 初始化一个可训练的参数矩阵  $N^{(0)} \in \mathbb{R}^{|\mathcal{E}| \times d^{(0)}}$  作为 **R-GCN** 的输入。随后对第  $l$  层的节点  $e$ , 使用如下式 (1) 来更新各节点表示:

$$n_e^{(l+1)} = \sigma \left( \sum_{r \in \mathcal{R}} \sum_{e' \in \mathcal{E}_e^r} \frac{1}{Z_{e,r}} W_r^{(l)} n_{e'}^{(l)} + W^{(l)} n_e^{(l)} \right), \quad (1)$$

其中  $n_e^{(l+1)}$  为节点  $e$  在第  $l+1$  层的表示,  $\mathcal{E}_e^r$  为图中节点  $e$  在  $r$  关系下的邻居节点的集合,  $W_r^{(l)}$  是第  $l$  层中与边类型  $r$  相关的可训练权重矩阵,  $W^{(l)}$  是第  $l$  层中与边类型  $r$  无关的可训练权重矩阵。  $Z_{e,r}$  是归一化因子, 例如可选定为  $|\mathcal{E}_e^r|$ 。在每一层, 每个节点都根据周围节点和边类型的信息来更新自己的向量表示, 在最后第  $L$  层得到了最终的实体表示矩阵  $N \in \mathbb{R}^{|\mathcal{E}| \times d}$ , 其每一个行向量为对应实体的向量表示。

### 4.2 融合多对话信息建模

获得实体的向量表示之后, 下一步是从用户的自然语言表达中提取必要的物品和非物品实体, 并融合历史对话数据与目标对话数据, 供后续对用户进行建模。

实体可以根据其性质分为两类, 一类称为物品, 即直接推荐给用户的实体, 例如在电影推荐场景中, 电影本身就是物品。另一类实体为非物品实体, 例如表 1 中的粗体字词汇“科幻片”、“克里斯托弗·诺兰”就是非物品实体。在用户表达中出现的这两类实体都能够为推荐任务带来有价值的信息。

目标对话  $\mathcal{C}^{(l)}$  中出现的所有实体构成上下文实体集  $\mathcal{E}_c \subset \mathcal{E}$ , 历史对话序列  $\mathcal{H}$  中出现的所有物品实

体构成历史物品集  $\mathcal{E}_h \subset \mathcal{E}$ 。进一步地, 依据这些实体可以从  $\mathcal{N}$  中取出相应的实体向量, 构成上下文实体矩阵  $\mathbf{M}_c \in \mathbb{R}^{|\mathcal{E}_c| \times d}$  和历史物品矩阵  $\mathbf{M}_h \in \mathbb{R}^{|\mathcal{E}_h| \times d}$ 。前者反映了用户在目标对话中展现的兴趣偏好, 后者则体现了用户在历史对话过程中体现出来的长期偏好。

为避免历史信息的冗余对结果造成负面影响, 本文使用注意力机制来甄别历史对话数据中对当前任务有价值的信息。根据历史物品与上下文实体的相关性, 对历史物品矩阵  $\mathbf{M}_h$  进行重组。首先根据如下式(2)来计算  $\mathbf{M}_h$  对  $\mathbf{M}_c$  的注意力分布矩阵:

$$\mathbf{A}_{hc} = \text{softmax}(\mathbf{M}_c \mathbf{W}_a \mathbf{M}_h^T), \quad (2)$$

满足  $\mathbf{A}_{hc} \in \mathbb{R}^{|\mathcal{E}_c| \times |\mathcal{E}_h|}$ 。这里使用了双线性模型注意力打分函数<sup>[28]</sup>来判定两个实体向量之间的相关性, 式中  $\mathbf{W}_a \in \mathbb{R}^{d \times d}$  为双线性模型的权重矩阵。获得注意力分布矩阵之后, 可以得到重组后的历史物品矩阵:

$$\mathbf{M}_{hc} = \mathbf{A}_{hc} \mathbf{M}_h \in \mathbb{R}^{|\mathcal{E}_c| \times d}. \quad (3)$$

$\mathbf{M}_c$  和  $\mathbf{M}_{hc}$  分别包含了用户在目标与历史对话中所展现出的偏好信息, 它们能够用来对用户进行更精细的个性化建模, 从而在物品数据库中寻找与用户最为匹配的推荐结果。

### 4.3 用户偏好建模与推荐

通过将用户表示成与实体向量相同维度的、处于同一语义空间的用户向量, 能够直接使用向量点积比较用户与物品实体的相似性, 从而向用户提供更加匹配的推荐结果。

这启示我们可以将  $\mathbf{M}_c$  和  $\mathbf{M}_{hc}$  从两个矩阵线性地压缩为一个  $d$  维的向量来代表用户, 由于这两个矩阵中的向量原本就来源于表示实体向量的语义空间, 那么经过线性压缩后得到的用户向量, 也自然而然地与该语义空间相匹配。

这种具有线性特征的压缩手段可以是直接对矩阵中的各向量取平均值, 但是本文采用了更复杂的自注意力机制<sup>[11]</sup>来完成这一步操作。

定义一个自注意力编码器  $\mathcal{S}: \mathbb{R}^{* \times d} \rightarrow \mathbb{R}^d$ , 这是一个从矩阵到向量的映射, 其中的矩阵可以由任意多个  $d$  维向量堆叠而成。编码器的具体形式为:

$$\mathcal{S}(\mathbf{M}) = \text{softmax}(\mathbf{w}_{a2} \tanh(\mathbf{W}_{a1} \mathbf{M}^T)) \times \mathbf{M}, \quad (4)$$

其中,  $\mathbf{W}_{a1} \in \mathbb{R}^{d_a \times d}$  是可训练的权重矩阵,  $\mathbf{w}_{a2} \in \mathbb{R}^{d_a}$  是可训练的权重向量。由此得到的自注意力编码器能够将任意多个实体向量, 按照一定的自注意力权

重比例线性组合成具备相当表达能力的用户向量。

为了权衡历史对话数据与目标对话数据之间的重要程度, 避免目标对话中的信息被丰富的历史对话数据所淹没, 本文依据“目标对话为主, 历史数据为辅”的原则, 采用层次化自注意力编码结构对两类信息进行编码。

首先对修正后的历史物品矩阵  $\mathbf{M}_{hc}$  进行编码, 得到与历史有关的用户向量表示  $\mathbf{t}_h = \mathcal{S}(\mathbf{M}_{hc}) \in \mathbb{R}^d$ , 再将其与上下文实体矩阵堆叠进一步得到用户矩阵  $\mathbf{M}_u = [\mathbf{t}_h; \mathbf{M}_c] \in \mathbb{R}^{(|\mathcal{E}_c|+1) \times d}$ 。对  $\mathbf{M}_u$  使用自注意力编码器编码得到最终的用户向量表示  $\mathbf{t}_u = \mathcal{S}(\mathbf{M}_u) \in \mathbb{R}^d$ 。层次化自注意力编码结构的设计体现了对两类数据重要性的权衡, 相当于先验知识的输入, 限制了历史信息发挥作用的空间。

最后通过对比用户与实体的相似性来进行推荐。具体来说, 通过向量的内积操作来得到各实体与用户的相关性打分, 并使用归一化函数  $\text{softmax}$  来获得物品实体最终出现的概率:

$$\mathbf{P}_{item} = \text{softmax}(\text{mask}(\mathbf{Nt}_u)) \in \mathbb{R}^{|\mathcal{I}|}, \quad (5)$$

其中的  $\text{mask}$  操作将所有非物品实体的打分置为  $-\infty$ , 避免向用户推荐非物品实体。

### 4.4 用户信息增强的对话生成

本文使用 Transformer<sup>[29]</sup>来完成系统的对话生成任务。这是一类能够对序列信息进行建模的编码器-解码器模型框架, 其中编码器能够对输入的词向量序列进行编码得到序列的高维表示, 然后将其输入解码器解码得到输出的词向量序列。输出的词向量再经过一层仿射变换分类器得到单词表上的概率分布。具体来说, 令  $\mathbf{o} \in \mathbb{R}^{d_w}$  表示输出的词向量,  $d_w$  为词向量维度, 最终得到的句子中词汇概率分布为  $\text{softmax}(\mathbf{W}\mathbf{o} + \mathbf{b}) \in \mathbb{R}^{|\mathcal{V}|}$ , 其中  $|\mathcal{V}|$  为词表大小,  $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times d_w}$  和  $\mathbf{b} \in \mathbb{R}^{|\mathcal{V}|}$  分别为权重矩阵和向量。

为了使生成的对话与当前推荐任务和用户偏好更为契合, 本文采用与 KBRD<sup>[6]</sup>相一致的做法, 将用户向量转化为词汇偏置, 再叠加到词概率分布的计算公式中。形式化地讲, 定义映射  $\mathcal{F}: \mathbb{R}^{d_w} \rightarrow \mathbb{R}^{|\mathcal{V}|}$ , 它能够将用户向量  $\mathbf{t}_u$  转化为词偏置  $\mathbf{b}_u = \mathcal{F}(\mathbf{t}_u)$ , 最终的词汇概率分布如下所示:

$$\mathbf{P}_{vocab} = \text{softmax}(\mathbf{W}\mathbf{o} + \mathbf{b} + \mathbf{b}_u) \in \mathbb{R}^{|\mathcal{V}|}. \quad (6)$$

至此, 一个端到端的个性化对话推荐算法框架 PCR 构建完成。推荐模块能够综合利用用户的历史对话数据和目标对话数据, 通过注意力机制和层次

化的编码权衡和利用两类信息, 完成对用户偏好的精细建模; 对话模块能够利用推荐模块构建的用户向量表示来生成更加符合用户偏好的回复信息。

## 5 实验

这一部分将详细介绍与实验相关的细节, 包括数据集的选用、不同任务评测指标的确定、系统实现的具体设定、所采用的基线模型类别, 以及对比实验的结果。

### 5.1 数据集

本文使用数据集 ReDial<sup>[5]</sup>来对模型进行训练和评测, 这是一个电影推荐场景下的对话推荐数据集, 标注者们在一定的指导下进行以电影推荐为主题的对话, 从而生成相应的数据。经过预处理后, 数据集总共有 10347 个对话, 每个对话都有相应的电影寻求者和推荐者, 这些对话分别属于 738 位电影寻求者, 其中 90% 以上的寻求者都有 5 个及以上的对话记录, 为完成个性化对话推荐任务提供了数据基础。

为了缓解推荐中可能遇到的冷启动问题, 并为模型构建可靠的实体向量表示, 本文使用外部知识库 DBpedia<sup>[26]</sup>来构建模型所需的图结构。在完成对话数据集到知识库的实体链接任务后, 共计得到 64362 个实体, 其中的电影实体有 6924 个。

### 5.2 评测指标

本文采用不同的指标来分别对推荐任务和对话任务进行评测。对于推荐任务, 使用 Hit@K、MRR@K、NDCG@K 来评价系统的表现, 分别反映了推荐结果的命中准确度和排序准确度。对于对话任务, 使用 BLEU<sup>[30]</sup>指标来评价生成的对话文本与目标的相似性, 使用 Distinct<sup>[31]</sup>指标来衡量生成结果的多样性, 以避免生成信息量不大、各处适用的文本。

### 5.3 实验设定

本文使用 CRSLab<sup>[8]</sup>框架来实现模型, 这是一个基于 PyTorch 实现的, 用于构建对话推荐系统的开源工具包, 其提供了易于使用的数据接口和大规模的标准评测机制, 便于对模型进行全方位的评测。

在推荐模块, 将实体向量和用户向量的维度设置为 128 维, R-GCN 模块的层数设置为 1, 归一化因子  $Z_{e,r}$  设定为  $|\mathcal{E}_e'|$ 。训练时的批处理大小设置为 2048, 使用 Adam<sup>[30]</sup>优化器进行优化, 学习率定为 0.003。

在对话模块, 将词向量的维度定为 300 维, 其 Transformer 模块编码器、解码器设置为 2 头自注意

力, 层数为 2。从用户向量转化为词偏置的映射  $\mathcal{F}$  定义为一个两层的前馈神经网络, 其中间层的大小为 512 维。训练时的批处理大小为 32, 同样使用 Adam 优化器进行优化, 学习率设定为 0.001。

### 5.4 对比实验结果

由于本文主要关注对话推荐系统中的推荐模块, 故选用如下四个模型作为基线模型进行比较, 来验证 PCR 模型在推荐任务上的高效性。

(1) Popularity: 将物品在数据集中出现的次数视为其受欢迎程度, 根据物品的受欢迎程度向用户进行推荐。

(2) AutoRec<sup>[31]</sup>: 基于自编码器的协同过滤方法。

(3) TextCNN<sup>[32]</sup>: 使用基于 CNN 的模型抽取文本特征来构建用户向量表示。

(4) KBRD<sup>[6]</sup>: 一个端到端的对话推荐算法模型, 其推荐模块结构如图 1 右侧所示, 它在构建用户向量时只使用了上下文实体进行预测。而本文的方法则考虑了历史对话中与用户发生交互的物品。

其中 Popularity、AutoRec 和 TextCNN 为推荐模型, 而 KBRD 同时具备处理推荐和对话任务的能力。

下面将列出实验结果, 包括在推荐任务和对话任务中, 本文的方法与基线模型效果的比较。

表 2 推荐任务评测

Table 2 Results on recommendation task			
模型	Hit@50	MRR@50	NDCG@50
Popularity	0.1831	0.0278	0.0575
AutoRec	0.2548	0.0532	0.0930
TextCNN	0.2944	0.0591	0.1059
KBRD	0.3961	0.0866	0.1492
PCR	<b>0.4121</b>	<b>0.0888</b>	<b>0.1539</b>

在推荐任务上的实验结果如表 2 所示, 本文使用 Hit@50、MRR@50、NDCG@50 三个指标来评价模型的推荐结果, 第一个指标反映推荐结果的命中程度, 后两个指标反映推荐结果的排序精度。可见在三个指标上, 本文的方法相对于基线模型均有所提升, 且显著性统计检验( $t$ -test)结果表明, 本文工作相对于最佳基线模型 KBRD 提升效果明显( $p$ -value < 0.05)。

## 6 讨论

本节将对模型结构的设计、数据集的可靠性和超参数的选取进行讨论。



## 6.1 模型结构设计

模型的结构实现中有两处最为关键的设计, 一是如何衡量历史对话数据与当前任务的关联程度, 也就是说应该采用什么样的注意力打分函数来评估历史物品的权重; 二是如何权衡历史对话数据与目标对话数据的重要性, 本文采用的方法是进行层次化编码。

在注意力机制中, 通过注意力打分函数对向量之间的匹配程度进行打分, 从而得到注意力分布权重。表 3 展示了常见的注意力打分函数, 其中  $x$  为输入向量,  $q$  为查询向量,  $D$  为向量的维度,  $W$  为双线性模型打分函数中的可训练权重矩阵。

表 3 常见的注意力打分函数

Table 3 Common attention scoring functions

类型	表达式
缩放点积	$\frac{x^T q}{\sqrt{D}}$
双线性	$x^T W q$

本文工作中注意力打分函数的选择并不是一个由经验支配的决定, 而是由实验结果所指引的。为了判断选择何种表达式能够更高效地提取历史对话数据中有用的信息, 仅使用历史物品进行推荐结果的预测。图 2 展示了采用不同的注意力打分函数完成这一任务时所取得的评测结果。可以发现, 使用双线性模型时所取得的结果最好。

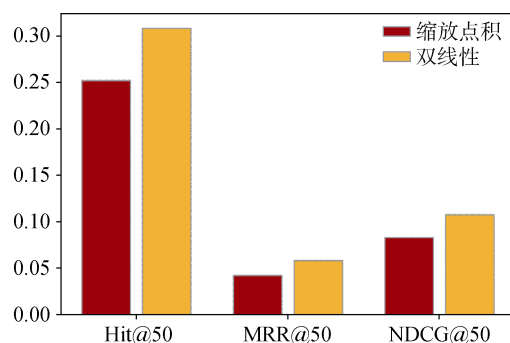


图 2 不同注意力打分函数的表现

Figure 2 Performances against different attention scoring functions

在权衡历史和目标对话数据的重要程度时, 本文采取的原则是“目标对话为主, 历史数据为辅”, 并设计了层次化自注意力编码结构对各类信息进行整合, 这种设计也来源于实验结果的指引。表 4 展示了使用层次化结构和非层次化结构(也就是历史物品和上下文实体直接进行整合)两种方法时, 模型在推

荐任务上的表现。

表 4 不同编码结构的性能表现

Table 4 Performances against different encoding structure

组别	Hit@50	MRR@50	NDCG@50
层次化结构	0.4121	0.0888	0.1539
非层次化结构	0.3970	0.0851	0.1481

可见在使用层次化编码结构时, 模型能够更精确地权衡各类信息之间的重要性, 从而得到更准确的推荐结果。

## 6.2 数据集的可靠性

事实上, 已有的对话推荐系统<sup>[5-7]</sup>假设数据集中的对话数据是独立同分布的, 而个性化对话推荐任务则要求对话数据是对于用户的条件分布。只有满足后一条件的数据集才能够证实本文工作的有效性, 否则本文实验结果中的效果提升可能只是一种假阳性效应。

由于本文所采用的 ReDial<sup>[5]</sup>数据集是由数据标注平台的工人生成的数据, 而非普通用户在不受干预的日常生活环境下生成的数据, 因此, 对话中的信息是否真实地反映了用户内在的兴趣偏好是值得怀疑的。本文设计了一个实验来验证 ReDial 数据集中的数据满足对于用户的条件分布。

实验中, 仅使用历史对话数据来预测用户可能喜欢的影片。在对照组中, 保持历史对话数据不变, 完全打乱它们对应的目标对话, 再进行预测。如果对话数据符合独立同分布的假设, 则对照组相对于实验组的结果不会有明显的差别; 如果对话数据符合对用户的条件分布, 则对照组的结果相对于实验组应该有明显的下降, 这是因为历史对话数据与目标对话数据所属的用户不一致。

表 5 数据集可靠性验证结果

Table 5 Verification results of dataset reliability

组别	Hit@50	MRR@50	NDCG@50
实验组	0.3178	0.0595	0.1107
对照组	0.2246	0.0382	0.0742

验证实验的结果如表 5 所示。对照组的结果相对于实验组发生了明显的下降, 因此数据集可以用于个性化对话推荐任务。实验也间接地验证了本文方法在基于多对话进行用户建模时的有效性。

## 6.3 历史对话数量选取

在实际应用场景中, 由于用户的历史对话数量

可能非常多,无法将所有的数据都加以利用。因此,只能选取一定数量以内的历史对话进行利用。

历史对话数量太少可能导致相关信息匮乏,数量过多会增加系统的计算负担,甚至可能带来一定的噪音影响。为了确定实验所选用的历史对话数量,需要将其作为变量对系统进行测试,寻找合适的拐点值作为本文实验所采用的值。

图 3 展示了 ReDial 数据集中用户的对话数量分布,展现出明显的长尾效应,大多数用户的对话数量都在 50 以内,个别用户的对话数量超过了 100 个。基于该统计结果,设置历史对话数量从 5~100,间隔为 5,实验测试不同历史对话数对系统推荐结果的影响。为了避免目标对话信息造成影响,仅使用历史对话数据进行预测,结果如图 4 所示。

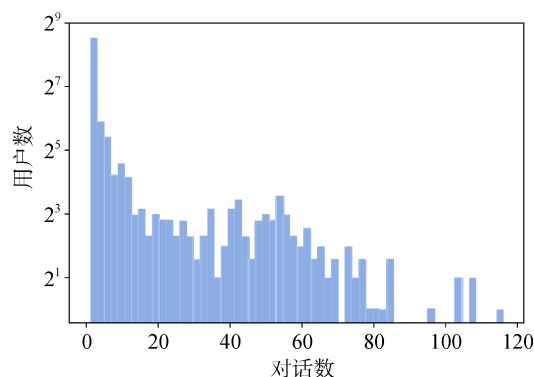


图 3 用户的对话数量分布

Figure 3 Distribution of user conversation numbers

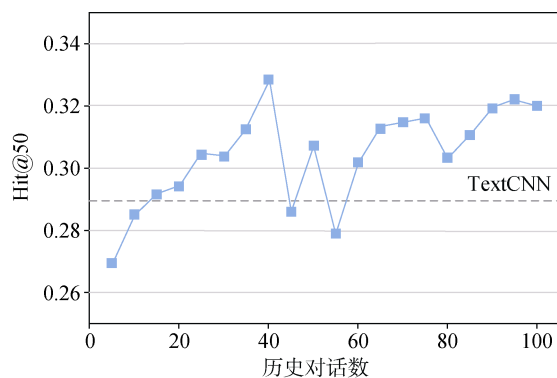


图 4 历史对话数对推荐结果的影响

Figure 4 Performances against different historical conversation numbers

历史对话数小于 40 时,推荐任务评测指标与历史对话数呈现正相关关系;在历史对话数超过 40 之后,推荐任务评测指标呈现不稳定的波动状态。前半部分的结果是符合直觉的,当历史对话数量越多,能够被利用的历史信息越充足,对推荐结果的提升作用也越明显。继续增加历史对话数,信息在变得充

足的同时也变得冗余,甚至引入噪音,导致推荐结果的准确度在一定范围内波动。据此,本文的实验选取历史对话数量为 40。

## 7 总结

本文提出了一个面向用户偏好建模的个性化对话推荐算法框架 **PCR**,它能够甄别用户历史对话数据中有价值的信息,通过层次化的自注意力编码结构得到用户向量表示,从而完成对候选物品的排序与推荐。实验结果表明,本文方法得到的结果在推荐任务和对话任务上相对于基线模型均有所提升。

接下来的工作中,可以进一步从历史对话数据中挖掘与用户偏好相关的信息,例如利用更普遍和丰富的文本信息对用户进行建模;同时引入更丰富的外部信息,缓解冷启动问题,并为用户提供更具多样性的推荐结果。

## 参考文献

- [1] He X R, Pan J F, Jin O, et al. Practical Lessons from Predicting Clicks on Ads At Facebook[C]. *The Eighth International Workshop on Data Mining for Online Advertising*, 2014: 1-9.
- [2] Zhou G R, Mou N, Fan Y, et al. Deep Interest Evolution Network for Click-through Rate Prediction[J]. *The AAAI Conference on Artificial Intelligence*, 2019, 33: 5941-5948.
- [3] Okura S, Tagami Y, Ono S, et al. Embedding-Based News Recommendation for Millions of Users[C]. *The 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017: 1933-1942.
- [4] Sarwar B, Karypis G, Konstan J, et al. Item-Based Collaborative Filtering Recommendation Algorithms[C]. *The tenth International Conference on World Wide Web - WWW'01*, 2001: 285-295.
- [5] Raymond Li, Samira Kahou, Hannes Schulz, et al. Towards Deep Conversational Recommendations[C]. *Advances in Neural Information Processing Systems*, 2018: 9748-9758.
- [6] Chen Q B, Lin J Y, Zhang Y C, et al. Towards Knowledge-Based Recommender Dialog System[C]. *The 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019: 1803-1813.
- [7] Zhou K, Zhao W X, Bian S Q, et al. Improving Conversational Recommender Systems via Knowledge Graph Based Semantic Fusion[C]. *The 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020: 1006-1014.
- [8] Kun Zhou, Xiaolei Wang, Yuanhang Zhou, et al. CRSLab: An Open-Source Toolkit for Building Conversational Recommender System[C]. *The 59th Annual Meeting of the Association for Com-*



- putational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, 2021: 185-193.
- [9] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, et al. Methods and metrics for cold-start recommendations[C]. *The 25th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002: 253-260.
- [10] Wenqiang Lei, Gangyi Zhang, Xiangnan He, et al. Interactive Path Reasoning on Graph for Conversational Recommendation[C]. *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020: 2073-2083.
- [11] Lin Z H, Feng M W, Santos C N D, et al. A Structured Self-Attentive Sentence Embedding[EB/OL]. 2017
- [12] Resnick P, Iacovou N, Suchak M, et al. GroupLens: An Open Architecture for Collaborative Filtering of Netnews[C]. *The 1994 ACM Conference on Computer Supported Cooperative Work*, 1994: 175-186.
- [13] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. *Computer*, 2009, 42(8): 30-37.
- [14] Richardson M, Dominowska E, Ragno R. Predicting Clicks: Estimating the Click-through Rate for New Ads[C]. *The 16th International Conference on World Wide Web - WWW'07*, 2007: 521-530.
- [15] Rendle S. Factorization Machines[C]. *2010 IEEE International Conference on Data Mining*, 2010: 995-1000.
- [16] Wang H, Wang N Y, Yeung D Y. Collaborative Deep Learning for Recommender Systems[C]. *The 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015: 1235-1244.
- [17] He X N, Liao L Z, Zhang H W, et al. Neural Collaborative Filtering[C]. *The 26th International Conference on World Wide Web*, 2017: 173-182.
- [18] Ying R, He R N, Chen K F, et al. Graph Convolutional Neural Networks for Web-Scale Recommender Systems[C]. *The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018: 974-983.
- [19] Yu X, Ren X, Sun Y Z, et al. Personalized Entity Recommendation: A Heterogeneous Information Network Approach[C]. *The 7th ACM International Conference on Web Search and Data Mining*, 2014: 283-292.
- [20] Zhang F Z, Yuan N J, Lian D F, et al. Collaborative Knowledge Base Embedding for Recommender Systems[C]. *The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016: 353-362.
- [21] Wang H W, Zhang F Z, Wang J L, et al. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems[C]. *The 27th ACM International Conference on Information and Knowledge Management*, 2018: 417-426.
- [22] Jamali M, Ester M. A Matrix Factorization Technique with Trust Propagation for Recommendation In Social Networks[C]. *The fourth ACM Conference on Recommender Systems*, 2010: 135-142.
- [23] Yueming Sun and Yi Zhang. Conversational recommender system[C]. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018: 235-244.
- [24] Yongfeng Zhang, Xu Chen, Qingyao Ai, et al. Towards conversational search and recommendation: System ask, user respond[C]. *The 27th ACM International Conference on Information and Knowledge Management*, 2018: 177-186.
- [25] Lei W Q, He X N, Miao Y S, et al. Estimation-Action-Reflection: Towards Deep Interaction between Conversational and Recommender Systems[C]. *The 13th International Conference on Web Search and Data Mining*, 2020: 304-312.
- [26] Lehmann J, Isele R, Jakob M, et al. DBpedia – a Large-Scale, Multilingual Knowledge Base Extracted from Wikipedia[J]. *Semantic Web*, 2015, 6(2): 167-195.
- [27] Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling Relational Data with Graph Convolutional Networks [C]. *European Semantic Web Conference*, 2018: 593-607.
- [28] Qiu X P. *Neural Networks and Deep Learning*[M]. Beijing: China Machine Press, 2020: 194-198.  
(邱锡鹏. 神经网络与深度学习[M]. 北京: 机械工业出版社, 2020: 194-198.)
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need[C]. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 2017: 6000-6010.
- [30] Kingma D, Ba J. Adam: A Method for Stochastic Optimization[EB/OL]. 2014.
- [31] Sedhain S, Menon A K, Sanner S, et al. AutoRec: Autoencoders Meet Collaborative Filtering[C]. *The 24th International Conference on World Wide Web*, 2015: 111-112.
- [32] Kim Y. Convolutional Neural Networks for Sentence Classification[C]. *The 2014 Conference on Empirical Methods in Natural Language Processing*, 2014: 1746-1751.



**尚琛展** 现于华中科技大学计算机科学与技术专业攻读学士学位。研究领域为推荐系统与自然语言处理。研究兴趣包括: 对话推荐系统、知识图谱等。Email: czshang@outlook.com



**赵鑫** 于 2014 年在北京大学获得博士学位。现任中国人民大学高瓴人工智能学院院长聘副教授(教授职称)。研究方向为复杂社交媒体环境下的个性化推荐算法与技术。Email: batmanfly@gmail.com