RecInDial: A Unified Framework for Conversational Recommendation with Pretrained Language Models

Lingzhi Wang[1,2]*, **Huang Hu**[4], **Lei Sha**[3], **Can Xu**[4], **Kam-Fai Wong**[1,2], **Daxin Jiang**[4]†
[1]The Chinese University of Hong Kong, Hong Kong, China
[2]MoE Key Laboratory of High Confidence Software Technologies, China
[3]University of Oxford, United Kingdom
[4]Microsoft Corporation, Beijing, China
[1,2]{lzwang,kfwong}@se.cuhk.edu.hk; [3]lei.sha@cs.ox.ac.uk;
[4]{huahu,caxu,djiang}@microsoft.com

Abstract

Conversational Recommender System (CRS), which aims to recommend high-quality items to users through interactive conversations, has gained great research interest recently. A CRS is usually composed of a recommendation module and a generation module. In the previous work, these two modules are loosely connected in the model training and are shallowly integrated during inference, where a simple switching or copy mechanism is adopted to incorporate recommended items into generated responses. Moreover, the current end-to-end neural models trained on small crowd-sourcing datasets (e.g., 10K dialogs in the ReDial dataset) tend to overfit and have poor chit-chat ability. In this work, we propose a novel unified framework that integrates recommendation into the dialog (*RecInDial*[1]) generation by introducing a vocabulary pointer. To tackle the low-resource issue in CRS, we finetune the large-scale pretrained language models to generate fluent and diverse responses, and introduce a knowledge-aware bias learned from an entity-oriented knowledge graph to enhance the recommendation performance. Furthermore, we propose to evaluate the CRS models in an end-to-end manner, which can reflect the overall performance of the entire system rather than the performance of individual modules, compared to the separate evaluations of the two modules used in previous work. Experiments on the benchmark dataset ReDial show our RecInDial model significantly surpasses the state-of-the-art methods. More extensive analyses show the effectiveness of our model.

1 Introduction

In recent years, there have been fast-growing research interests to address Conversational Recommender System (CRS) (Li et al., 2018; Sun and Zhang, 2018; Zhou et al., 2020a), due to the booming of intelligent agents in e-commerce platforms. It aims to recommend target items to users through interactive conversations. Traditional recommender systems perform personalized recommendations based on user's previous implicit feedback like clicking or purchasing histories, while CRS can proactively ask clarification questions and extract user preferences from conversation history to conduct precise recommendations. Existing generative methods (Chen et al., 2019; Zhou et al., 2020a; Ma et al., 2020; Liang et al., 2021) are generally composed of two modules, *i.e.*, a recommender module to predict precise items and a dialogue module to generate free-form natural responses containing the recommended items. Such methods usually utilize Copy Mechanism (Gu et al., 2016) or Pointer Network (Gulcehre et al., 2016) to inject the recommended items into the generated replies. However, these strategies cannot always incorporate the recommended items into the generated responses precisely and appropriately. On the other hand, most of the existing CRS datasets (Li et al., 2018; Zhou et al., 2020b; Liu et al., 2020, 2021) are relatively small (∼10K dialogues) due to the expensive crowd-sourcing labor. The end-to-end neural models trained on these datasets from scratch are prone to be overfitting and have undesirable quality on the generated replies in practice.

Encouraged by the compelling performance of pre-training techniques, we present a pre-trained language models (PLMs) based framework called *RecInDial* to address these challenges. *RecInDial* integrates the item recommendation into the dialogue generation under the pretrain-finetune schema. Specifically, RecInDial finetunes the powerful PLMs like DialoGPT (Zhang et al., 2020) together with a Relational Graph Convolutional Network (RGCN) to encode the node representation of an item-oriented knowledge graph. The former aims to generate fluent and diverse dialogue

*Work performed during internship at Microsoft STCA.
†Corresponding author: djiang@microsoft.com.
[1]The code is available at https://github.com/Lingzhi-WANG/PLM-BasedCRS

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 489–500
November 20–23, 2022.

...
User: That sounds good. I could go with a classic. Have you seen Troll 2 (1990)? I'm looking for a horrible movie. cheesy horror
Human: Tuesday 13, you like? *ReDial*: Black Panther (2018) is a good one too. *KBRD*: or It (2017) *KGSF*: I would recommend watching it. *OUR*: yes I have seen that one. It was good. I also liked the movie It (2017).
...

Table 1: A conversation example with movies recommendation from the test set of ReDial dataset.

responses based on the strong language generation ability of PLMs, while the latter is to facilitate the item recommendation by learning better structural node representations. To bridge the gap between response generation and item recommendation, we expand the generation vocabulary of PLMs to include an extra item vocabulary. Then a vocabulary pointer is introduced to control when to predict a target item from the item vocabulary or a word from the ordinary vocabulary in the generation process. The introduced item vocabulary and vocabulary pointer effectively unify the two individual processes of response generation and item recommendation into one single framework in a more consistent fashion.

To better illustrate the motivation of our work, Table 1 shows a conversation example on looking for horrible movies and the corresponding replies generated by four models (*ReDial* (Li et al., 2018), *KBRD* (Chen et al., 2019), *KGSF* (Zhou et al., 2020a), OUR) together with the ground truth reply in the corpus (Human). As we can see, the previous work tends to generate short (e.g., "KBRD: or It (2017)") or in-coherent responses (e.g., "KGSF: I would recommend watching it."), which is resulted from the overfitting on the small dataset as we mentioned before. Different from them, our model can generate more informative and coherent sentences which shows a better chatting ability. In additon, we can notice that KGSF fails to raise a recommendation in the response "I would recommend watching it" ("it" should be replaced with a specific item name in a successful combination of generation and recommendation results), which is probably due to the insufficient semantic knowledge learned and an ineffective copy mechanism. Our proposed unified PLM-based framework with a vocabulary pointer can effectively solve the issue.

Furthermore, to better investigate the end-to-end CRS system, we argue to evaluate the performance of recommendation by checking whether the final responses contain the target items. Existing works separately evaluate the performance of the two modules, *i.e.*, dialogue generation and item recommendation. However, a copy mechanism or pointer network cannot always inject the recommended items into generated replies precisely and appropriately as we mentioned before. The performance of the final recommendations is actually lower than that of the recommender module. For instance, the Recall@1 of the recommender module in KGSF (Zhou et al., 2020a) is 3.9% while the actual performance is only 0.9% when evaluating the final integrated responses (see Table 3).

We conduct extensive experiments on the popular benchmark REDIAL (Li et al., 2018). Our RecInDial model achieves a remarkable improvement on the recommendation over the state-of-the-art, and the generated responses are also significantly better on automatic metrics as well as human evaluation. Further ablation studies and quantitative and qualitative analyses demonstrate the superior performance of our approach.

The contributions of this work can be:

- We propose a PLM-based framework called RecInDial for conversational recommendation. RecInDial finetunes the large-scale PLMs together with a Relational Graph Convolutional Network to address the low-resource challenge in the current CRS.
- By introducing an extra item vocabulary with a vocabulary pointer, RecInDial effectively unifies two components of item recommendation and response generation into a PLM-based framework.
- Extensive experiments show RecInDial significantly outperforms the state-of-the-art methods on the evaluation of both dialogue generation and recommendation.

2 Related Work

Existing works in CRS can be mainly divided into two categories, namely attribute-based CRS and open-ended CRS.

Attribute-based CRS. The attribute-based CRS can be viewed as a question-driven task-oriented dialogue system (Zhang et al., 2018; Sun and Zhang, 2018). This kind of system proactively asks clarification questions about the item attributes to infer user preferences, and thus search for the optimal candidates to recommend. There are various ask-

ing strategies studied by existing works, such as entropy-ranking based approach (Wu et al., 2018), generalized binary search based approaches (Zou and Kanoulas, 2019; Zou et al., 2020), reinforcement learning based approaches (Chen et al., 2018; Lei et al., 2020a; Deng et al., 2021), adversarial learning based approach (Ren et al., 2020b) and graph based approaches (Xu et al., 2020; Lei et al., 2020b; Ren et al., 2021; Xu et al., 2021). Another line of research on this direction address the trade-off issue between exploration (*i.e.*, asking questions) and exploitation (*i.e.*, making recommendations) to achieve both the engaging conversations and successful recommendations, especially for the cold-start users. Some of them leverage bandit online recommendation methods to address cold-start scenarios (Li et al., 2010, 2016b; Christakopoulou et al., 2016; Li et al., 2020), while others focus on the asking strategy with fewer turns (Lei et al., 2020a,b; Shi et al., 2019; Sun and Zhang, 2018).

Open-ended CRS. Existing works (Li et al., 2018; Lei et al., 2018; Jiang et al., 2019; Ren et al., 2020a; Hayati et al., 2020; Ma et al., 2020; Liu et al., 2020; Wang et al., 2022) on this direction explore CRS through more free-form conversations, including proactively asking clarification questions, chatting with users, providing the recommendation, etc. Multiple datasets have been released to help push forward the research in this area, such as REDIAL (Li et al., 2018), TG-REDIAL (Chinese) (Zhou et al., 2020b), INSPIRED (Hayati et al., 2020) and DuRecDial (Liu et al., 2020, 2021). Li et al. (2018) make the first attempt on this direction and contribute the benchmark dataset REDIAL by the paired crowd-workers (*i.e.*, Seeker and Recommender). Follow-up studies (Chen et al., 2019; Zhou et al., 2020a,b) leverage the multiple external knowledge to enhance the performance of open-ended CRS. CR-Walker (Ma et al., 2020) is proposed to perform the tree-structured reasoning on the knowledge graph to introduce relevant items, while MGCG (Liu et al., 2020) addresses the transition policy from a non-recommendation dialogue to a recommendation-oriented one. Besides, Zhou et al. (2021) develop an open-source toolkit CRSLab to further facilitate the research on this direction. Most of these works utilize pointer network (Gulcehre et al., 2016) or copy mechanism (Gu et al., 2016; Sha et al., 2018) to inject the recommended items into generated replies. Our work lies in the research of open-ended CRS. While different from the previous work, we present a PLM-based framework for CRS, which finetunes the large-scale PLMs together with a pre-trained Relational Graph Convolutional Network (RGCN) to address the low-resource challenge in CRS.

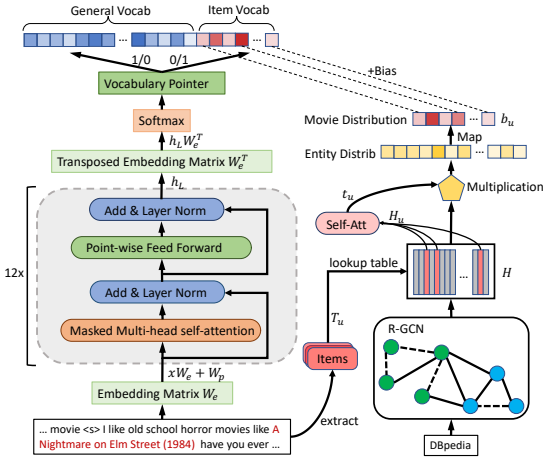


Figure 1: Model overview of RecInDial.

Another line of related work lies in the end-to-end task-oriented dialogs (Wu et al., 2019; He et al., 2020; Raghu et al., 2021), which also require response generation based on a knowledge base but not for recommendations.

3 Methodology

In this section, we present our proposed RecInDial model. Figure 1 shows the model overview. We first formalize the conversational recommendation task and then detail our PLM-based response generation module together with the vocabulary pointer. After that, we introduce how to incorporate the knowledge from an item-oriented knowledge graph with an RGCN into the model. Finally, we describe the model training objectives.

3.1 Problem Formalization

The input of a CRS model contains the history context of a conversation, which is denoted as a sequence of utterances $\{t_1, t_2, ..., t_m\}$ in chronological order (m represents the number of utterances). Each utterance is either given by the seeker (user) or recommender (the model), which contains the token sequence $\{w_{i,1}, w_{i,2}, ..., w_{i,n_i}\}$ ($1 \leq i \leq m$), where w_{ij} is the j-th token in the i-th utterance and n_i is the number of tokens in i-th utterance. Note that we define the name of an item as a single token and do not tokenize it. The output token sequence by the model is denoted as $\{w_{n+1}, w_{n+2}, ..., w_{n+k}\}$, where k is the number of generated tokens and $n = \sum_1^m n_i$ is the total num-

ber of tokens in context. When the model conducts the recommendation, it will generate an item token w_{n+i} $(1 \leq i \leq k)$ together with the corresponding context. In this way, recommendation item and response are generated concurrently.

3.2 Response Generation Model

In this subsection, we introduce how to extend PLMs to handle CRS task and produce items recommendation during the dialogue generation.

PLM-based Response Generation. Given the input (*i.e.*, the conversation history context $\{t_1, t_2, ..., t_m\}$), we concatenate the history utterances into the context $C = \{w_1, w_2, ..., w_n\}$ where n is the total number of tokens in the context. Then the probability of the generated response $R = \{w_{n+1}, w_{n+2}, ..., w_{n+k}\}$ is formulated as:

$$\mathrm{PLM}(R|C) = \prod_{i=n+1}^{n+k} p(w_i|w_1, ..., w_{i-1}). \quad (1)$$

where $\mathrm{PLM}(\cdot|\cdot)$ denotes the PLMs of Transformer (Vaswani et al., 2017) architecture. For a multi-turn conversation, we can construct N such context-response pairs, where N is the number of utterances by the recommender. Then we finetune the PLMs on all possible (C, R) pairs constructed from the dialogue corpus. By this means, not only does our model inherit the strong language generation ability of the PLMs, but also simultaneously can learn how to generate the recommendation utterances on the relatively small CRS dataset.

PLM-based Item Generation. To integrate the item recommendation into the generation process of PLMs, we propose to expand the generation vocabulary of PLMs by including an extra item vocabulary. We devise a vocabulary pointer to control when to generate tokens from the ordinary vocabulary or from the item vocabulary. Concretely, we regard an item as a single token and add all items into the item vocabulary. Hence, our model can learn the relationship between context words and candidate items. Such a process integrates the response generation and item recommendation into a unified model that can perform the end-to-end recommendation through dialogue generation.

Vocabulary Pointer. We first preprocess the dialogue corpus and introduce two special tokens `[RecS]` and `[RecE]` to indicate the start and end positions of the item in utterance. Then we divide the whole vocabulary V into V_G and V_R, where

Algorithm 1 Vocabulary Pointer based Generation for RecInDial

Input: history context C, general and item vocabulary V_G, V_R
Output: generated response R

```
extract appeared entities from C as user preference T_u
compute knowledge-aware bias b_u based on T_u using Eq. 5 to 8
R ← {}
n ← 0
I_vp ← 0, V ← V_G
while n < N_max do
    w_n = Decode(C ⋃ R, V, b_u)   ▷ Generate w_n based on the previous tokens and bias from V
    R ← R ⋃ {w_n}
    if w_n = [RecS] then          ▷ Generate tokens from V_R
        I_vp ← 1, V ← V_R
    else if w_n = [RecE] then     ▷ Generate tokens from V_G
        I_vp ← 0, V ← V_G
    else if w_n = [EOS] then      ▷ Generation is done
        break
    end if
    n ← n + 1
end while
return R
```

V_G includes the general tokens (*i.e.*, tokens in the original vocabulary of PLM) and `[RecS]` while V_R contains the all item tokens and `[RecE]`. We then introduce a binary *Vocabulary Pointer* I_{vp} to guide the generation from V_G or V_R. The model generates tokens in V_G when $I_{vp} = 0$, and generates the tokens in V_R when $I_{vp} = 1$, which can be formulated as follows:

$$p(w = w_i) = \frac{exp(\phi_I(w_i) + \tilde{h}_i)}{\sum_{w_j \in V} exp(\phi_I(w_j) + \tilde{h}_j)} \quad (2)$$

$$\phi_I(w_j) = \begin{cases} 0, & I_{vp} = 0, w_j \in V_G \text{ or} \\ & I_{vp} = 1, w_j \in V_R, \\ -inf, & I_{vp} = 1, w_j \in V_G \text{ or} \\ & I_{vp} = 0, w_j \in V_R \end{cases}, \quad (3)$$

where $\tilde{h} = h_L W_e^T$ is the feature vector before the softmax layer in Figure 1, $\tilde{h}_i$ means the feature value of the i-th token. I_{vp} is initialized as 0 at the beginning of the generation and won't change until the model produces `[RecS]` or `[RecE]`. It changes to 1 if the model produces `[RecS]` (*i.e.*, the model begins to generate items) and changes back to 0 if `[RecE]` is emitted. Such a procedure continues until the turn is finished. With the *Vocabulary Pointer*, our model can alternatively switch between generating response words and recommending items based on its previous outputs in a unified fashion.

To help readers better understand the Vocabulary Pointer mechanism, we summarize the process in Algorithm 1.

3.3 Knowledge Graph Enhanced Finetuning

Due to the difficulty of fully understanding user preferences by the conversation context, it is necessary to introduce the external knowledge to encode the user preferences when finetuning response generation model. Inspired by the previous work (Chen et al., 2019; Zhou et al., 2020a), we also employ a knowledge graph from DBpedia (Lehmann et al., 2015) and perform entity linking (Daiber et al., 2013) to the items in the dataset, which helps better model the user preferences. A triple in DBpedia is denoted by $< e_1, r, e_2 >$, where $e_1, e_2 \in \mathcal{E}$ are items or entities from the entity set $\mathcal{E}$ and r is entity relation from the relation set $\mathcal{R}$.

Relational Graph Propagation. We utilize R-GCN (Schlichtkrull et al., 2018) to encode structural and relational information in the knowledge graph to entity hidden representations. Formally, the representation of node e at $(l+1)$-th layer is:

$$\boldsymbol{h}_e^{(l+1)} = \sigma(\sum_{r\in\mathcal{R}}\sum_{e'\in\mathcal{E}_e^r}\frac{1}{Z_{e,r}}\boldsymbol{W}_r^{(l)}\boldsymbol{h}_{e'}^{(l)} + \boldsymbol{W}^{(l)}\boldsymbol{h}_e^{(l)}), \quad (4)$$

where $\boldsymbol{h}_e^{(l)} \in \mathbb{R}^{d_E}$ is the node representation of e at the l-th layer, and $\mathcal{E}_e^r$ denotes the set of neighboring nodes for e under the relation r. $\boldsymbol{W}_r^{(l)}$ is a learnable relation-specific transformation matrix for the embedding from neighboring nodes with relation r, while $\boldsymbol{W}^{(l)}$ is another learnable matrix for transforming the representations of nodes at the l-th layer and $Z_{e,r}$ is a normalization factor.

At the last layer L, structural and relational information is encoded into the entity representation $\boldsymbol{h}_e^{(L)}$ for each $e \in \mathcal{E}$. The resulting knowledge-enhanced hidden representation matrix for entities in $\mathcal{E}$ is denoted as $\boldsymbol{H}^{(L)} \in \mathbb{R}^{|\mathcal{E}|\times d_E}$. We omit the (L) in the following paragraphs for simplicity.

Entity Attention. Given a conversation context, we first collect the entities appeared in the context, and then we represent the user preference as $\mathcal{T}_u = e_1, e_2, ..., e_{|\mathcal{T}_u|}$, where $e_i \in \mathcal{E}$. After looking up the knowledge-enhanced representation table of entities in $\mathcal{T}_u$ from $\boldsymbol{H}$, we get:

$$\boldsymbol{H}_u = (\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_{|\mathcal{T}_u|}), \quad (5)$$

where $\boldsymbol{h}_i \in \mathbb{R}^{d_E}$ is the hidden vector of entity e_i. Then the self-attention mechanism (Lin et al., 2017) is applied to $\boldsymbol{H}_u$, which outputs a distribution α_u over $|\mathcal{T}_u|$ vectors:

$$\alpha_u = softmax(\boldsymbol{w}_{a2}tanh(\boldsymbol{W}_{a1}\boldsymbol{H}_u^T)), \quad (6)$$

where $\boldsymbol{W}_{a1} \in \mathbb{R}^{d_a\times d_E}$ and $\boldsymbol{w}_{a2} \in \mathbb{R}^{1\times d_a}$ are learnable parameters. Then we get the final representation for user history u as follows:

$$\boldsymbol{t}_u = \alpha_u \boldsymbol{H}_u. \quad (7)$$

Knowledge-Aware Bias. To incorporate the knowledge from the constructed knowledge graph into our model while generating recommendation items, we first map the derived user representation $\boldsymbol{t}_u$ into the item vocabulary space $|V_R|$ as follows:

$$\boldsymbol{b}_u = \boldsymbol{t}_u \boldsymbol{H}^T \boldsymbol{M}_b, \quad (8)$$

where $\boldsymbol{M}_b \in \mathbb{R}^{|\mathcal{E}|\times|V_R|}$ are learnable parameters. Then we add b_u to the projection outputs before softmax operation in the generation as a bias. In this way, our model can produce items in aware of their relational knowledge and thus enhance the performance of recommendation.

3.4 Recommendation in Beam Search

To embed the top-k item recommendation into the generation, we develop a revised beam search decoding. Specifically, when we finish the generation for one response, we first check whether it contains the item names (i.e., whether it generates recommendations). If yes, then we choose the top-k items between `[RecS]` and `[RecE]` according to the probability scores at current time-step.

3.5 Learning Objectives

There are two objectives, *i.e.*, node representation learning on knowledge graph and the finetuning of response generation model. For the former, we optimize the R-GCN and the self-attention network based on the cross entropy of item prediction:

$$\mathcal{L}_{kg} = \sum_{(u,i)\in\mathcal{D}_1} -log(\frac{exp(\boldsymbol{t}_u\boldsymbol{H}^T)_i}{\sum_j exp(\boldsymbol{t}_u\boldsymbol{H}^T)_j}), \quad (9)$$

where the item i is the ground-truth item and u is the corresponding user history, while $\mathcal{D}_1$ contains all training instances and $\boldsymbol{t}_u\boldsymbol{H}^T \in \mathbb{R}^{|\mathcal{E}|}$.

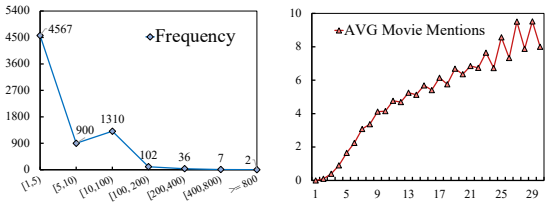
For the latter, we optimize another cross entropy loss for all generated responses, denoted as R. The following formula summarizes the process:

$$\mathcal{L}_{gen} = \sum_{(C,R)\in\mathcal{D}_2}\sum_{w_i\in R} -log(p(w_i|w_{<i}, C)), \quad (10)$$

where $p(w_i)$ refers to Eq. 2 and $\mathcal{D}_2$ contains all (C, R) pairs constructed from the dataset. We train the whole model end-to-end with the joint effects of the two objectives $\mathcal{L}_{kg} + \mathcal{L}_{gen}$.

Conversations		Movies	
# of convs	10006	# of mentions	51699
# of utterances	182150	# of movies	6924
# of users	956	avg mentions	7.5
avg token length	6.8	max mentions	1024
avg turn #	18.2	min mentions	1

Table 2: Statistics of ReDial dataset. "#" means number and "avg" refers to average.



(a) Movie # Distribution (b) Position Distribution

Figure 2: For Figure 2(a), X-axis: the movie mentions range; Y-axis: movie numbers. For Figure 2(b), X-axis: turn positions; Y-axis: average movie mentions.

4 Experimental Setup

Datasets. We evaluate our model on the benchmark dataset REDIAL (Li et al., 2018). Due to the collection difficulty of the real world data, most the previous work (Li et al., 2018; Chen et al., 2019; Zhou et al., 2020a) only conducts experiments on this single dataset. The statistics of REDIAL dataset is shown in Table 2. Detailed statistics of movie mentions are shown in Figure 2(a). Most of the movies occur less than 5 times in the dataset, which indicates an obvious data imbalance problem in the REDIAL. We also show the relationship between the average number of movie mentions and the number of dialog turns in Figure 2(b). As we can see, there are less than 2 movie mentions when the dialogue turn number is less than 5. Finally, we follow (Li et al., 2018) to split the dataset into 80-10-10, for training, validation and test.

Parameter Setting. We finetune the small size pre-trained DialoGPT model[2], which consists of 12 transformer layers. The dimension of embeddings is 768. It is trained on 147M multi-turn dialogues from Reddit discussion threads. For the knowledge graph (KG), both the entity embedding size and the hidden representation size are set to 128, and we set the layer number for R-GCN to 1. For BART baseline, we finetune the base model [3] with 6 layers in each of the encoder and decoder, and a hidden size of 1024. For GPT-2 baseline, we finetune the small model[4]. For all model's training, we adopt Adam optimizer and the learning rate is chosen from $\{1e{-}5, 1e{-}4\}$. The batch size is chosen from {32, 64}, the gradient accumulation step is set to 8, and the warm-up step is chosen from {500, 800, 1000}. All the hyper-parameters are determined by grid-search.

[2] https://huggingface.co/microsoft/DialoGPT-small

[3] https://huggingface.co/facebook/bart-base

[4] https://huggingface.co/gpt2

Baselines and Comparisons. We first introduce two baselines for recommender and dialogue modules, respectively. (1) **Popularity**. It ranks the movie items according to their historical frequency in the training set without a dialogue module. (2) **Transformer** (Vaswani et al., 2017). It utilizes a transformer-based encoder-decoder to generate responses without recommender module.

We then compare the following baseline models in the experiment: (3) **ReDial** (Li et al., 2018). It consists of a dialogue generation module based on HRED (Serban et al., 2017), a recommender module based on auto-encoder (He et al., 2017), and a sentiment analysis module. (4) **KBRD** (Chen et al., 2019). It utilizes a knowledge graph from DBpedia to model the relational knowledge of contextual items or entities, and the dialogue generation module is based on the transformer architecture. (5) **KGSF** (Zhou et al., 2020a). It incorporates and fuses both word-level and entity-level knowledge graphs to learn better semantic representations for user preferences. (6) **GPT-2**. We directly finetune GPT-2 and expand its vocabulary to include the item vocabulary. (7) **BART**. We directly finetune BART and expand its vocabulary to include the same item vocabulary. (8) **DialoGPT**. We directly finetune DialoGPT and expand its vocabulary to include same item vocabulary.

For our RecInDial, in addition to the full model (9) **RecInDial**, we also evaluate two variants: (10) **RecInDial *w/o* VP**, where we remove the vocabulary pointer; and (11) **RecInDial *w/o* KG**, where the knowledge graph part is removed.

Evaluation Metrics. As we discussed above, the previous works evaluate the recommender and dialogue modules separately. Following the previous setting (Chen et al., 2019; Zhou et al., 2020a), we evaluate the recommender module by Recall@k (k = 1, 10, 50). Besides, we also evaluate Recall@k in an end-to-end manner, *i.e.*, to check whether the

final produced response contains the target item. In such a setting, the Recall@K score not only depends on whether the ground truth item appears in the top K recommendation list but also reply on if the recommended item is successfully injected into the generated sentences. Therefore, the end-to-end evaluation is fair for all models and applicable for K = 1, 10, 50. For the dialogue module, automatic metrics include: (1) **Fluency**: perplexity (PPL) measures the confidence of the generated responses. (2) **Relevance**: BLEU-2/4 (Papineni et al., 2002) and Rouge-L (Lin, 2004). (3) **Diversity**: Distinct-n (Dist-n) (Li et al., 2016a) are defined as the number of distinct n-grams divided by the total amount of words. Specifically, we use Dist-2/3/4 at the sentence level to evaluate the diversity of generated responses. Besides, we also employ Item Ratio introduced in KGSF (Zhou et al., 2020a) to measure the ratio of items in the generated responses.

5 Experimental Results

In this section, we first report the comparison results on recommendation and response generation. Then we discuss the human evaluation results. After that, we show an example to illustrate how our model works, followed by qualitative analysis.

5.1 Results on Recommendation

The main experimental results for our RECINDIAL and baseline models on recommendation side are presented in Table 3. And we can draw several observations from the results.

There is a significant gap between the performance of the recommender module and the performance of the final integrated system. KGSF, the state-of-the-art model, achieves 3.9% Recall@1 in the recommender module evaluation but yields only 0.9% in the evaluation of the final produced responses. This indicates that the integration strategies utilized by previous methods have significant harm on the recommendation performance.

Finetuning PLMs on the small CRS dataset is effective. As we can see, compared to non-PLM based methods, directly finetuning GPT-2/BART/DialoGPT on the REDIAL achieves the obvious performance gain on recommendation.

Our RecInDial model significantly outperforms the SOTAs on recommendation performance. As shown in Table 2, our RecInDial achieves the best Recall@k (k = 1, 10, 50) scores under the end-to-end evaluation, which demonstrates the superior performance of the PLMs with the unified design.

Models	Eval on Rec Module			End-to-End Eval		
	R@1	R@10	R@50	R@1	R@10	R@50
Baselines						
Popularity	1.2	6.1	17.9	1.2	6.1	17.9
ReDial	2.4	14.0	32.0	0.7	4.4	10.0
KBRD	3.1	15.0	33.6	0.8	3.8	8.8
KGSF	3.9	18.3	37.8	0.9	4.2	8.8
GPT-2	-	-	-	1.4	6.5	14.4
BART	-	-	-	1.5	-	-
DialoGPT	-	-	-	1.7	7.1	13.8
RecInDial	-	-	-	**3.1**	**14.0**	**27.0**

Table 3: Main comparison results on recommendation. R@k refers to Recall@k. RecInDial outperms the baselines significantly ($p<0.01$, paired t-test).

Models	R@1	R@10	R@50	Item Ratio	BLEU	Rouge-L
RecInDial	**3.1**	**14.0**	**27.0**	**43.5**	**20.7**	**17.6**
RecInDial *w/o* VP	1.8	8.8	19.5	17.8	18.5	14.6
RecInDial *w/o* KG	2.3	9.4	20.1	39.8	17.7	12.9

Table 4: Comparison results on ablation study.

5.2 Results on Dialogue Generation

Since CRS aims to recommend items during natural conversations, we conduct both automatic and human evaluations to investigate the quality of generated responses by RecInDial and baselines.

Automatic Evaluation. Table 5 shows the main comparison results on Dist-2/3/4, BLEU-2/4, Rouge-L and PPL. As we can see, RecInDial significantly outperforms all baselines on Dist-n, which indicates that *PLM helps generate more diverse responses.* Previous works suffer from the low-resource issue due to the small crowd-sourcing CRS dataset and tend to generate boring and singular responses. On the other hand, *our RecInDial model tends to recommend items more frequently*, as the Item Ratio score of RecInDial is much higher than those of baselines. Besides, our RecInDial and PLM-based methods consistently achieve remarkable improvement over non-PLM based methods on all metrics, which demonstrates the superior performance of PLMs on dialogue generation.

Human Evaluation. To further investigate the effectiveness of RecInDial, we conduct a human evaluation experiment, where four crowd-workers are employed to score on 100 context-response pairs that are randomly sampled from the test set. Then, we collect the generation results of RecInDial and the baseline models and compare their performance on the following three aspects: (1) **Fluency**. Whether a response is organized in regular English grammar and easy to understand. (2) **Informativeness**. Whether a response is meaningful and not a "safe response", and repetitive

Models	Dist-2	Dist-3	Dist-4	IR	BL-2	BL-4	Rouge-L	PPL↓
Baselines								
Transformer	14.8	15.1	13.7	19.4	-	-	-	-
ReDial	22.5	23.6	22.8	15.8	17.8	7.4	16.9	61.7
KBRD	26.3	36.8	42.3	29.6	18.5	7.4	17.1	58.8
KGSF	28.9	43.4	51.9	32.5	16.4	7.4	14.3	131.1
GPT-2	35.4	48.6	44.1	14.5	17.1	7.7	11.3	56.3
BART	37.6	49.0	43.5	16.0	17.8	9.3	13.1	55.6
DialoGPT	47.6	55.9	48.6	15.9	16.7	7.8	12.3	56.0
RecInDial	**51.8**	**62.4**	**59.8**	**43.5**	**20.4**	**11.0**	**17.6**	**54.1**

Table 5: Automatic metrics on generated responses. IR denotes the Item Ratio.

Models	Fluency	Informative	Coherence	Kappa
HUMAN	1.93	1.70	1.69	0.80
ReDial	1.90	1.28	1.21	0.75
KBRD	1.92	1.32	1.26	0.78
KGSF	1.91	1.05	1.10	0.85
RecInDial	1.93	1.65	1.60	0.84

Table 6: Human evaluation results.

responses are regarded as uninformative. (3) **Coherence**. Whether a response is coherent with the previous context. The crowd-workers give a score on the scale of [0, 1, 2] to show the quality of the responses, and higher scores indicate better qualities.

We calculate the average score for each model, as well as the ground truth that humans give. As shown in Table 6, our model shows better performance than all the baselines. Interestingly, ground-truth Human cannot get a 100% correctness in all the four evaluation metrics. The reason may be that words and phrases sent by human annotators on AMT platform sometimes are the casual usage popular on Internet, which has the wrong grammar. For the fluency, all models generate fluent utterances and show similar performance. For the informativeness, our RecInDial achieves better performance than the baselines, which indicates RecInDial tends to generate more meaningful responses.

5.3 Ablation Study

We then report the performance comparisons on RecInDial's variants. Table 4 shows the end-to-end recommendation performance and generation results. Removing the vocabulary pointer leads to significant drops on R@k and Item Ratio. *This indicates Vocabulary Pointer (VP) introduced in RecInDial is crucial to the performance of item recommendation.* The reason is that the generation process would lose the guidance to switch between general tokens and recommended items without the help of the vocabulary pointer. Besides, we can find that *knowledge graph enhanced finetuning helps achieve better recommendation performance.* Introducing the node representations learned on the knowledge graph can model the user preference better, which could further enhance the recommendation performance.

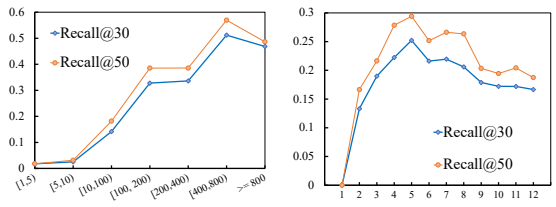
5.4 Qualitative Analysis

In this subsection, we present a conversation example to illustrate how our model works in practice.

Conversation Example	
User:	I like to watch scary movies
Human:	Tell me what kind of movie do you like? Did you watch *It (2017)*? And *It (1990)*?
KBRD:	Hi, what kind of movies do you like?
KGSF:	Hello!
RecInDial:	Have you seen *It (2017)*?
User:	I have seen the old one
Human:	Did you watch *The Exorcist (1973)*?
KBRD:	*Get Out (2017)*
KGSF:	I would recommend *Get Out (2017)*
RecInDial:	How about *Psycho (1960)*?
User:	I have not seen the new one I do like the exorcist
Human:	Ok, you should watch the new one even though I would rather the old one. And did you watch *The Last Exorcism (2010)*? I hope I have helped
KBRD:	I haven't seen that one
KGSF:	I would recommend it
RecInDial:	Yes, it was good. I hope you enjoy it. Have a good day. Bye

Table 7: A conversation example on *movie item* recommendation. The responses of **KBRD**, **KGSF** and **RecInDial** are from the test results for corresponding models. **Human** responses are ground-truth.

In Table 7, the *Seeker* states that he likes scary movies. Our model successfully captured the keyword of "scary" and recommends a famous scary movie "*It (2017)*" while the state-of-the-art model KGSF produces a safe response "Hello!", which shows our RecInDial can generate the responses that are more coherent with the context. Interestingly, after the *Seeker* says he watched the old "*It (1990)*", our model recommends another horror movie "*Psycho (1960)*" also released in the last century. The possible reason is that RecInDial infers the seeker is interested in old horror movies. The example in Table 7 shows that our RecInDial tends to generate a more informative response than KGSF. In addition, we find that KGSF always generates "I would recommend *Item*" (*Item* is replaced with *Get out (2017)* in this example) and "I would recommend it.". The first response pattern successfully integrates the movie item into the response,



(a) Recall over Mentions (b) Recall over Turn #

Figure 3: Y-axis: Recall. For Fig. 3(a), X-axis: Movie mentions range. For Fig. 3(b), X-axis: turn numbers.

while the second fails to make a complete recommendation, which reveals the drawback of the copy mechanism in KGSF.

5.5 Further Analysis

Analysis on Data Imbalance. As we discussed aforementioned, the movie occurrence frequency shows an imbalanced distribution over different movies (see Figure 2(a)). To investigate the effect, we report the Recall@30 and Recall@50 scores over movie mentioned times in Figure 3(a). As we can see, the recall scores for low-frequency movies (with mentioned times less than 10) are much lower than those high-frequency movies (with $>$ 100 mentions). However, most of the movies (5467 out of 6924 movies) in the REDIAL dataset are low-frequency movies, which leads to relatively low results in the overall performance.

Analysis on Cold Start. REDIAL dataset suffers from the cold-start problem. It is hard for models to recommend precise items in the first few turns of the conversation. We report the Recall@30 and Recall@50 scores of our RecInDial over different dialogue turns in Figure 3(b). Generally, we can see that the recall scores are getting better with richer information gradually obtained from dialogue interactions. The scores begin to drop when there are more than 5 turns. The possible reason is that as the conversation goes deeper, the Seekers are no longer satisfied with the recommended high-frequency movies but prefer more personalized recommendations, which makes it more difficult to predict in practice.

6 Conclusion

This paper presents a novel unified PLM-based framework called *RecInDial* for CRS, which integrates the item recommendation into the generation process. Specifically, we finetune the large-scale PLMs together with a relational graph convolutional network on an item-oriented knowledge graph. Besides, we design a vocabulary pointer mechanism to unify the response generation and item recommendation into the existing PLMs. Extensive experiments on the CRS benchmark dataset REDIAL show that RecInDial significantly outperforms the state-of-the-art methods.

Acknowledgements

We would like to thank the anonymous reviewers for their feedback and suggestions. The research described in this paper is partially supported by HKSAR ITF No. ITT/018/22LP.

References

Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards knowledge-based recommender dialog system. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1803–1813, Hong Kong, China. Association for Computational Linguistics.

Yihong Chen, Bei Chen, Xuguang Duan, Jian-Guang Lou, Yue Wang, Wenwu Zhu, and Yong Cao. 2018. Learning-to-ask: Knowledge acquisition via 20 questions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1216–1225. ACM.

Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 815–824. ACM.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 121–124.

Yang Deng, Yaliang Li, Fei Sun, Bolin Ding, and Wai Lam. 2021. Unified conversational recommendation policy learning via graph-based reinforcement learning. *arXiv preprint arXiv:2105.09710*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149, Berlin, Germany. Association for Computational Linguistics.

Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan Shi, and Zhou Yu. 2020. INSPIRED: Toward sociable recommendation dialog systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8142–8152, Online. Association for Computational Linguistics.

Junhua He, Hankz Hankui Zhuo, and Jarvan Law. 2017. Distributed-representation based hybrid recommender system with short item descriptions. *arXiv preprint arXiv:1703.04854*.

Zhenhao He, Yuhong He, Qingyao Wu, and Jian Chen. 2020. Fg2seq: Effectively encoding knowledge for end-to-end task-oriented dialog. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8029–8033. IEEE.

Shaojie Jiang, Pengjie Ren, Christof Monz, and Maarten de Rijke. 2019. Improving neural response diversity with frequency-aware cross-entropy loss. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2879–2885. ACM.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020a. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 304–312. ACM.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Melbourne, Australia. Association for Computational Linguistics.

Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020b. Interactive path reasoning on graph for conversational recommendation. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 2073–2083. ACM.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 661–670. ACM.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9748–9758.

Shijun Li, Wenqiang Lei, Qingyun Wu, Xiangnan He, Peng Jiang, and Tat-Seng Chua. 2020. Seamlessly unifying attributes and items: Conversational recommendation for cold-start users. *arXiv preprint arXiv:2005.12979*.

Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. 2016b. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 539–548. ACM.

Zujie Liang, Huang Hu, Can Xu, Jian Miao, Yingying He, Yining Chen, Xiubo Geng, Fan Liang, and Daxin Jiang. 2021. Learning neural templates for recommender dialogue system. *arXiv preprint arXiv:2109.12302*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. 2021. Durecdial 2.0: A bilingual parallel corpus for conversational recommendation. *arXiv preprint arXiv:2109.08877*.

Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Towards conversational recommendation over multi-type dialogs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1036–1049, Online. Association for Computational Linguistics.

Wenchang Ma, Ryuichi Takanobu, Minghao Tu, and Minlie Huang. 2020. Bridging the gap between conversational reasoning and interactive recommendation. *arXiv preprint arXiv:2010.10333*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Dinesh Raghu, Atishya Jain, Sachindra Joshi, et al. 2021. Constraint based knowledge base distillation in end-to-end task oriented dialogs. *arXiv preprint arXiv:2109.07396*.

Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020a. Thinking globally, acting locally: Distantly supervised global-to-local knowledge selection for background based conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8697–8704.

Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Zi Huang, and Kai Zheng. 2021. Learning to ask appropriate questions in conversational recommendation. *arXiv preprint arXiv:2105.04774*.

Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Nguyen Quoc Viet Hung, Zi Huang, and Xiangliang Zhang. 2020b. Crsal: Conversational recommender systems with adversarial learning. *ACM Transactions on Information Systems (TOIS)*, 38(4):1–40.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3295–3301. AAAI Press.

Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2018. Order-planning neural text generation from structured data. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5414–5421. AAAI Press.

Chen Shi, Qi Chen, Lei Sha, Hui Xue, Sujian Li, Lintao Zhang, and Houfeng Wang. 2019. We know what you will ask: A dialogue system for multi-intent switch and prediction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 93–104. Springer.

Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 235–244. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Lingzhi Wang, Shafiq Joty, Wei Gao, Xingshan Zeng, and Kam-Fai Wong. 2022. Improving conversational recommender system via contextual and time-aware modeling with less domain-specific knowledge. *arXiv preprint arXiv:2209.11386*.

Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019. Global-to-local memory pointer networks for task-oriented dialogue. *arXiv preprint arXiv:1901.04713*.

Xianchao Wu, Huang Hu, Momo Klyen, Kyohei Tomita, and Zhan Chen. 2018. Q20: Rinna riddles your mind by asking 20 questions. *Japan NLP*.

Hu Xu, Seungwhan Moon, Honglei Liu, Bing Liu, Pararth Shah, Bing Liu, and Philip Yu. 2020. User memory reasoning for conversational recommendation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5288–5308, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kerui Xu, Jingxuan Yang, Jun Xu, Sheng Gao, Jun Guo, and Ji-Rong Wen. 2021. Adapting user preference to online feedback in multi-round conversational recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 364–372.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 177–186. ACM.

Kun Zhou, Xiaolei Wang, Yuanhang Zhou, Chenzhan Shang, Yuan Cheng, Wayne Xin Zhao, Yaliang Li, and Ji-Rong Wen. 2021. CRSLab: An open-source toolkit for building conversational recommender system. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 185–193, Online. Association for Computational Linguistics.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020a. Improving conversational recommender systems via knowledge graph based semantic fusion. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1006–1014. ACM.

Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020b. Towards topic-guided conversational recommender system. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4128–4139, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jie Zou, Yifan Chen, and Evangelos Kanoulas. 2020. Towards question-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 881–890. ACM.

Jie Zou and Evangelos Kanoulas. 2019. Learning to ask: Question-based sequential bayesian product search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 369–378. ACM.