RID: A Unified Framework for Conversational Recommender Systems with Pretrained Language Models

Anonymous ACL submission

Abstract

Conversational Recommender Systems (CRS), which aim to recommend high-quality items to users through interactive conversations, have gained great research attention 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 network or copy mechanism is adopted to incorporate recommended items into generated responses. Moreover, the current end-to-end neural models trained on small crowdsourcing datasets (e.g., 10K dialogs in the ReDial dataset) tend to be overfitting and have poor chit-chat ability. In this work, we propose a novel unified framework called RID that integrates recommendation into the dialog generation by introducing a vocabulary pointer. To tackle the low-resource issue in CRS, we fine-tune the large-scale pretrained language model 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 histories 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 RID to address these challenges. RID integrates the item Recommendation into the Dialogue generation under the pretrain-finetune schema. Specifically, RID 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 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., 2020), 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 incoherent 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 addition, 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 RID 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 RID for conversational recommendation. RID 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, RID effectively unifies two components of item recommendation and response generation into a PLM-based framework.
• Extensive experiments show RID 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 asking 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 addresses the trade-off 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) 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 RedDial 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 CRS@Lab 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.

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 RID 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 \( \{t_1, t_2, \ldots, t_m\} \) 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 \( \{w_{i,1}, w_{i,2}, \ldots, w_{i,n_i}\} \) (\( 1 \leq i \leq m \)), 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 \( \{w_{n+1}, w_{n+2}, \ldots, w_{n+k}\} \), where \( k \) is the number of generated tokens and \( n = \sum_i n_i \) is the total number 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:

\[
\text{PLM}(R|C) = \prod_{i=n+1}^{n+k} p(w_i|w_1, ..., w_{i-1}). \tag{1}
\]

where \( \text{PLM}(\cdot|\cdot) \) 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 fine-tune 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 \([\text{RecS}]\) and \([\text{RecE}]\) 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 \( V_G \) includes the general tokens (i.e., tokens in the original vocabulary of PLM) and \([\text{RecS}]\) while \( V_R \) contains all the item tokens and \([\text{RecE}]\). 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:

\[
p(w = w_i) = \frac{\exp(\phi_I(w_i) + \tilde{h}_i)}{\sum_{j \in V} \exp(\phi_I(w_j) + \tilde{h}_j)} \tag{2}
\]

\[
\phi_I(w_j) = \begin{cases} 0, & I_{vp} = 0, w_j \in V_G \\ 1, & I_{vp} = 1, w_j \in V_R \\ \inf, & I_{vp} = 0, w_j \in V_R \end{cases} \tag{3}
\]

where \( \tilde{h} = h_L W_e^T \) is the feature vector before the softmax layer in Figure 1, \( \hat{h}_i \) 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 \([\text{RecS}]\) or \([\text{RecE}]\). It changes to 1 if the model produces \([\text{RecS}]\) (i.e., the model begins to generate items) and changes back to 0 if \([\text{RecE}]\) 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.

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 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 \(e_1, e_2 \in E\) are items or entities from the entity set \(E\) and \(r\) is entity relation from the relation set \(R\).

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:

\[
h_e^{(l+1)} = \sigma(\sum_{r \in E \cap E^r} \frac{1}{2\sigma_e} W_r^{(l)} h_e^{(l)} + W_e^{(l)} h_e^{(l)}), \tag{4}
\]

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

At the last layer $L$, structural and relational information is encoded into the entity representation $h_e^{(L)}$ for each $e \in E$. The resulting knowledge-enhanced hidden representation matrix for entities in $E$ is denoted as $H^{(L)} \in \mathbb{R}^{|E| \times d_e}$. 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, $e_1, e_2, ..., e_{|\mathcal{T}_u|}$, where $e_i \in \mathcal{E}$. After looking up the knowledge-enhanced representation table of entities in $\mathcal{T}_u$ from $H$, we get:

$$H_u = (h_1, h_2, ..., h_{|\mathcal{T}_u|}),$$

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

$$\alpha_u = \text{softmax}(w_{a1}\tanh(W_{a1}H_u^{(L)})),$$

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

$$t_u = \alpha_u H_u.$$  

**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 $t_u$ into the item vocabulary space $|V_R|$ as follows:

$$b_u = t_u H^T M_b,$$

where $M_b \in \mathbb{R}^{|E| \times |V_R|}$ 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 $[\text{RecS}]$ and $[\text{RecE}]$ 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:

$$\mathcal{L}_{kg} = \sum_{(u,i) \in \mathcal{D}_1} -\log(\frac{\exp(t_i H^T)}{\sum_j \exp(t_j H^T)}),$$

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 $t_i H^T \in \mathbb{R}^{|E|}$.

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

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

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 $\mathcal{L}_{kg} + \mathcal{L}_{gen}$.

### 4 Experimental Setup

**Datasets.** We evaluate our model on the benchmark dataset REDIAL (Li et al., 2018), which collects the human conversations on movie recommendation on Amazon Mechanical Turk (AMT) platform with pair crowd-workers (i.e., Seeker and Recommender). The statistics of REDIAL dataset is shown in Table 2. More data statistics can be found in Appendix. 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\(^1\), 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. More parameter settings can be found in Appendix.

**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

\(^1\)https://huggingface.co/microsoft/DialoGPT-small
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. (9) RID, where we remove the vocabulary pointer; and (11) RID 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) 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 RID 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 RID model significantly outperforms the SOTA on recommendation performance. As shown in Table 2, our RID 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.

#### 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 RID and baselines.

**Automatic Evaluation.** Table 5 shows the main comparison results on Dist-2/3/4, BLEU-2/4,
Whether a response is meaningful and not a “safe
response”, which demonstrates the superior performance of
our model shows better perfor-
mance 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 pop-
ular on Internet, which has the wrong grammar. For
the fluency, all models generate fluent utterances
and show similar performance. For the informativ-
eness, our RID achieves better performance than the
baselines, which indicates RID tends to generate
more meaningful responses.

5.3 Ablation Study

We then report the performance comparisons on
RID’s variants. Table 4 shows the end-to-end rec-
ommendation performance and generation results.

Removing the vocabulary pointer leads to signifi-
cant drops on R@k and Item Ratio. This indicates
Vocabulary Pointer (VP) introduced in RID is cru-
ial 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 vocab-
ulary pointer. Besides, we can find that knowledge
graph enhanced finetuning helps achieve better rec-
ommendation 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 exam-
ple to illustrate how our model works in practice.
More examples are included in Appendix.

In Figure 7, the Seeker states that he likes scary
movies. Our model successfully captured the key-

![Table 3: Main comparison results on recommendation.](image)

<table>
<thead>
<tr>
<th>Models</th>
<th>Eval on Rec Module</th>
<th>End-to-End Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R@1 R@10 R@50</td>
<td>R@1 R@10 R@50</td>
</tr>
<tr>
<td>Baselines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Popularity</td>
<td>1.2 6.1 17.9</td>
<td>1.2 6.1 17.9</td>
</tr>
<tr>
<td>ReDial</td>
<td>2.4 14.0 32.0</td>
<td>0.7 4.4 10.0</td>
</tr>
<tr>
<td>KBRD</td>
<td>3.1 15.0 33.6</td>
<td>0.8 3.8 8.8</td>
</tr>
<tr>
<td>KGFS</td>
<td>3.9 18.3 37.8</td>
<td>0.9 4.2 8.8</td>
</tr>
<tr>
<td>GPT-2</td>
<td>- - -</td>
<td>1.4 6.5 14.4</td>
</tr>
<tr>
<td>BART</td>
<td>- - 1.5</td>
<td>-</td>
</tr>
<tr>
<td>DiagloGPT</td>
<td>- - 1.7</td>
<td>7.1 13.8</td>
</tr>
<tr>
<td>RID</td>
<td>- - 3.1</td>
<td>14.0 27.0</td>
</tr>
</tbody>
</table>

Table 3: Main comparison results on recommendation.
R@k refers to Recall@k (k=1, 10 and 50).

![Table 4: Comparison results on ablation study.](image)

<table>
<thead>
<tr>
<th>Models</th>
<th>R@1 R@10 R@50 Item Ratio BLEU Rouge-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>RID w/o VP</td>
<td>3.1 14.0 27.0 43.5 20.7 17.6 19.7 14.6</td>
</tr>
<tr>
<td>RID w/o KG</td>
<td>2.3 9.4 19.1 17.8 18.5 17.7 12.9</td>
</tr>
</tbody>
</table>

Table 4: Comparison results on ablation study.

Rouge-L and PPL. As we can see, RID signifi-
cantly 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 RID model tends
to recommend items more frequently, as the
Item Ratio score of RID is much higher than those of baselines. Besides, our RID and PLM-based
methods consistently achieve remarkable improve-
ment 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 RID, we conduct a human evalu-
ation 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 RID and the
baseline models and compare their performance on
the following three aspects: (1) **Fluency.** Whether
a response is organized in regular English gram-
mar and easy to understand. (2) **Informativeness.**
Whether a response is meaningful and not a “safe
response”, and repetitive responses are regarded as uninformative. (3) **Coherence.** Whether a re-
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. More details in
human evaluation are provided in Appendix.

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 perfor-
mance 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 pop-
ular on Internet, which has the wrong grammar. For
the fluency, all models generate fluent utterances
and show similar performance. For the informativ-
eness, our RID achieves better performance than the
baselines, which indicates RID tends to generate
more meaningful responses.

![Table 5: Automatic metrics on generated responses.](image)

<table>
<thead>
<tr>
<th>Models</th>
<th>Dist-2 Dist-3 Dist-4</th>
<th>IR</th>
<th>BL-2 BL-4 Rouge-L PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>RID</td>
<td>51.8 62.4 59.8 45.3 20.4 11.0 17.6 54.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

![Table 6: Human evaluation results.](image)

| Models         | Fluency | Informative | Coherence | Kappa | |
|----------------|---------|-------------|-----------|-------|
| RID            | 1.93    | 1.70        | 1.69      | 0.80  |
| RID w/o VP     | 1.90    | 1.28        | 1.21      | 0.75  |
| RID w/o KG     | 1.91    | 1.05        | 1.10      | 0.85  |

Table 6: Human evaluation results.
Table 7: A conversation example on movie item recommendation. The responses of KBRD, KGSF and RID are from the test results for corresponding models. Human responses are ground-truth.

<table>
<thead>
<tr>
<th>User:</th>
<th>I like to watch scary movies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human:</td>
<td>Tell me what kind of movie do you like? Did you watch <em>It (2017)</em> and <em>It (1990)</em>?</td>
</tr>
<tr>
<td>KBRD:</td>
<td>Hi, what kind of movies do you like?</td>
</tr>
<tr>
<td>KGSF:</td>
<td>Hello!</td>
</tr>
<tr>
<td>RID:</td>
<td>Have you seen <em>It (2017)</em>?</td>
</tr>
</tbody>
</table>

User: | I have seen the old one |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Human:</td>
<td>Did you watch <em>The Exorcist (1973)</em>?</td>
</tr>
<tr>
<td>KBRD:</td>
<td><em>Get Out (2017)</em></td>
</tr>
<tr>
<td>KGSF:</td>
<td>I would recommend <em>Get Out (2017)</em></td>
</tr>
<tr>
<td>RID:</td>
<td>How about <em>Psycho (1960)</em>?</td>
</tr>
</tbody>
</table>

User: | I have not seen the new one I do like the exorcist |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Human:</td>
<td>Ok, you should watch the new one even though I would rather the old one. And did you watch <em>The Last Exorcism (2010)</em>? I hope I have helped</td>
</tr>
<tr>
<td>KBRD:</td>
<td>I haven’t seen that one</td>
</tr>
<tr>
<td>KGSF:</td>
<td>I would recommend it</td>
</tr>
<tr>
<td>RID:</td>
<td>Yes, it was good. I hope you enjoy it. Have a good day. Bye</td>
</tr>
</tbody>
</table>

...
References


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.


Wenqiang Lei, Xiangnan He, Yisong Mao, 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, Gangyi Zhang, Xiangnan He, Yisong Mao, 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.


Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal.


Appendix

A Limitations

In this section, we mainly talk about the limitation of single-reference evaluation problem in an open-ended conversation recommendation system and the limitation of our model when handling new items.

The Limitation of Single-Reference Evaluation

As you can see in Table 1 or Table 7, there is only one ground truth movie item for every utterance. Under such a setting, humans may even not suggest the item that is the same as the ground truth. Therefore, it may result in the Recall@K (K = 1, 10, 50) could not reflect the true performance of a system well. Besides, the sizes of the currently available datasets are quite small which exacerbates the impact of the single-reference evaluation.

The Limitation of Handling New Items. Our framework cannot handle items that are not existed in the training set, since we extend the vocabulary with static item set. However, handling new items that never appear during training is important when applying in a real-world application system. We might resolve it by making the item set dynamic. We leave it to our future work.

B Data Statistics and Parameter Setting

Data Statistics. Detailed statistics of movie mentions are shown in Figure 3(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 3(b). As we can see, there are less than 2 movie mentions when the dialogue turn number is less than 5.

Parameter Setting. For BART baseline, we finetune the base model 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. 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, the warm up step is chose from \{500, 800, 1000\}. All the hyper-parameters are determined by grid-search.

\(\text{https://huggingface.co/facebook/bart-base}\)

\(\text{https://huggingface.co/gpt2}\)

C Vocabulary Pointer Algorithm

To help readers better understand the Vocabulary Pointer mechanism described in Section 3.2, we summarize the process in Algorithm 1.

D Dialog Examples

We present some representative examples here to illustrate the characteristics of models.

Example 1. As we can see in Figure 4, there are following observations:
(1) KGSF ignores the need that the CRS model should say hello first. Therefore, the first generation result of KGSF is none.
(2) Our model tends to say much longer responses than the other three baseline models.
(3) The last line in Example 4, our model generates “Have you seen the new Jumanji (2017)?”, which means our model has captured that there is a series of Jumanji movies and then recommends a new one.

Example 2. As we can see in Figure 5, there are following observations:
(1) ReDIAL and KBRD always generate repetitive responses.
**Algorithm 1** Vocabulary Pointer based Generation for RecInDial

**Input:** history context $C$, general and item vocabulary $V_G$, $V_R$

**Output:** generated response $R$

1. extract appeared entities from $C$ as user preference $T_u$
2. compute knowledge-aware bias $b_u$ based on $T_u$ using Eq. 5 to 8
3. $R \leftarrow \{\}$
4. $n \leftarrow 0$
5. $I_{vp} \leftarrow 0$, $V \leftarrow V_G$

while $n < N_{max}$ do

6. $w_n = \text{Decode}(C \cup R, V, b_u)$ \hspace{1cm} $\triangleright$ Generate $w_n$
7. $R \leftarrow R \cup \{w_n\}$
8. if $w_n = \text{[RecS]}$ then \hspace{1cm} $\triangleright$ Start to generate tokens from $V_R$
9. \hspace{1cm} $I_{vp} \leftarrow 1$, $V \leftarrow V_R$
10. else if $w_n = \text{[RecE]}$ then \hspace{1cm} $\triangleright$ Start to generate tokens from $V_G$
11. \hspace{1cm} $I_{vp} \leftarrow 0$, $V \leftarrow V_G$
12. else if $w_n = \text{[EOS]}$ then \hspace{1cm} $\triangleright$ Generation is done

13. break
14. end if
15. $n \leftarrow n + 1$
16. end while

17. return $R$

---

(2) Our model has stronger ability of chit-chat. When the *Seeker* shows his affection for *Johnny Depp*, our model replies, “Yes, I love *Johnny Depp* too !”, which shows our model can generate coherent and human-like responses.

(3) Better evaluation method is needed. Our model generates “Have you ever seen *A Nightmare on Elm Street* (1984) ?”. Though the response generated differs from the *Human*, our mentioned movie item is proper according to the following conversation content. Therefore, how to better evaluate the dialog response and the item recommendation deserves more efforts.

### E Human Evaluation Details

**Evaluation Criterion.** Table 8 shows the scoring details for human evaluation experiment. As can be seen, we specifies the scoring details for each evaluation metric.
Table 8: Scoring details for human evaluation.

<table>
<thead>
<tr>
<th>Score</th>
<th>Fluency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The response has many grammar mistakes.</td>
</tr>
<tr>
<td></td>
<td>The response is hard to understand.</td>
</tr>
<tr>
<td>1</td>
<td>The response has minor grammar mistakes.</td>
</tr>
<tr>
<td></td>
<td>Some part of the response is hard to understand.</td>
</tr>
<tr>
<td>2</td>
<td>The response is in correct grammar and easy to understand.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Score</th>
<th>Coherence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The response is not related with the context.</td>
</tr>
<tr>
<td></td>
<td>The response simply repeats the context.</td>
</tr>
<tr>
<td></td>
<td>The response has obvious conflicts with the context.</td>
</tr>
<tr>
<td>1</td>
<td>The response has minor conflicts with the context.</td>
</tr>
<tr>
<td></td>
<td>There are some minor logic conflicts in the response.</td>
</tr>
<tr>
<td>2</td>
<td>The response is coherent with the context.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Score</th>
<th>Informativeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The response does not contain any information.</td>
</tr>
<tr>
<td></td>
<td>This response just repeats the context and fails to bring any additional information.</td>
</tr>
<tr>
<td></td>
<td>The information is invalid, as the coherence score is 0.</td>
</tr>
<tr>
<td>1</td>
<td>The information has conflicts with common sense.</td>
</tr>
<tr>
<td></td>
<td>There are factual errors in the response.</td>
</tr>
<tr>
<td>2</td>
<td>The response contains appropriate and correct information.</td>
</tr>
</tbody>
</table>

Seeker: I have seen The Conjuring (2013) I very much enjoyed this movie. When I was younger I liked movies like Friday the 13th (1980).
Human: I liked that one too!
Redial: i have n’t seen that one . i ’ve heard good things about it .
KBRD: i have n’t seen that one . i ’ve heard Halloween (1978) was good .
KGSF: How about The Descent (2005) ?
Our: Have you ever seen A Nightmare on Elm Street (1984) ?
Seeker: I also liked to watch the A Nightmare on Elm Street (1984) Series.
Human: I loved that as well mainly because Johnny Depp is in it
Redial: i have n’t seen that one yet . i ’ve seen It (2017)
KBRD: i have n’t seen that one . i have heard Halloween (1978) was good .
KGSF: I also recommend Insidious (2011)
Our: Have you ever seen The Conjuring (2013)? I liked that one too!
Seeker: I love Johnny Depp! So handsome, such a great actor. I love johnny Depp in Alice in Wonderland (2010).
Human: Me too one of my favorites from him is The Astronaut’s Wife (1999)
Redial: i have n’t seen that one yet . i ’ve seen It (2017) .
KBRD: i have n’t seen that one . i will check it out .
KGSF: I hope you enjoy them !
Our: Yes, I love Johnny Depp too! Have you seen the new It (2017) movie?

Figure 5: Second example with movies from the test set of ReDial.