Towards Policy-Guided Conversational Recommendation with Dialogue Acts

Anonymous ACL submission

Abstract

Conversation Recommender System (CRS) aims to recommend items through nature conversation. Existing works in open-ended CRS mainly focus on recommendation and generation, but lacks of control over dialogue policy. In addition, the system is unable to adapt user profile to the user's feedback. Thus, we present a new dataset named **DA-ReDial**[1] (Recommendation through Dialogue guided by Dialogue Act). We summarize 10 representative Dialog Acts and label dialogue with the DAs schema. To solve the problems above, we also propose a novel CRS called **PGCR**, which stands for Policy-Guided Conversational Recommendation. It is able to formulate a DA-aware user profile, leverage Dialogue Acts to explicitly model the discourse structure of conversation and better guide the response generation. Extensive experiments on the new dataset show that our proposed model outperforms most baselines in dialog generation and recommendation. Also, the Policy Network fine-tuned by self-play can better control the dialogue policy and contribute a lot to recommendation strategy and user engagement in conversation.

1 Introduction

Recently, Conversational Recommender System (CRS) has witnessed rapid development and gained much attention due to its research potential (Deng et al., 2021; Wang et al., 2021) and industrial values (Shum et al., 2018; Zhang et al., 2018a). Different from the mechanical recommendation system (Koren et al., 2009; Rendle, 2010) , CRS can recommend items for users via nature conversations. CRS, from the perspective of dialogue, can be divided into attribute-centric (Zhang et al., 2018a; Lei et al., 2020c; Zou et al., 2020) or open-ended (Chen et al., 2019; Liao et al., 2019; Liu et al., 2020; Hayati et al., 2020). Usually, both of the two categories consist of a recommender module and a conversation module. Recently, the attribute-centric CRS (Lei et al., 2020a,c) performs well with the aid of the Policy Module, which could design strategy and further guide the generation. Specially, the CRS is allowed to explicitly ask user's preference for item attributes or recommend a list of items at each turn (Lei et al., 2020c). Then the corresponding dialog template will be selected from the library (Christakopoulou et al., 2018; Zhang et al., 2018b; Deng et al., 2021) and generated with entities related with items. With the explicit guide from policy network, attribute-centric CRS demonstrates sharp control over conversation.

Label	Dialogue Acta (DAs)	Designed For
0	CHAT	BOT & SEEKER
1	YES NO QUESTION	BOT
2	WHAT QUESTION	BOT
3	RECOMMEND BY QUESTION	BOT
4	RECOMMEND BY STATEMENT	BOT
5	HINT BY QUESTION	SEEKER
6	HINT BY STATEMENT	SEEKER
7	ACCEPT	SEEKER
8	REJECT	SEEKER
9	NEUTRAL	SEEKER

Table 1: Dialogue Acts Schema. We abstract 10 kinds of dialogue acts commonly adopted in CRS.

However, the open-ended CRS puts more emphasis on the flexibility and fluency of natural conversation (Chen et al., 2019; Liao et al., 2019; Kang et al., 2019; Ma et al., 2020; Chen et al., 2020; Hayati et al., 2020; Zhang et al., 2021), thus showing little control over the policy strategy. Despite the utilization of switching network (Li et al., 2018) or CopyNet (Gu et al., 2016) during decoding, it can only exercise the word-level responses instead of the utterance-level. Furthermore, the attribute-centric CRS (Shum et al., 2018) is able to fully apprehend the user's feedback and adapt it to modeling user profile, while the open-ended CRS cannot take full advantage of the feedback, since it cannot extract the implicit policy in seeker's utterance.

[1]The dataset and code will be available on Github soon.

Among this background, we formulate a new open-end CRS dataset named **DA-ReDial** (Recommendations through Dialogue guided by Dialogue Act) in this work. Its backbone is based on ReDial (Li et al., 2018) and the formulation of DA-ReDial is simple. After an in-depth observation of the dialogue part and the prior works about dialogue acts (Takanobu et al., 2020; Ma et al., 2021), we design a high-quality Dialogue Acts schema, which can represent the acts of almost all conversations. As shown in table 1, the schema includes 10 kinds of dialogue acts, of which four are designed for bot, four for seeker and one for both. With this DAs schema, we weekly label all dialogs of ReDial.

In addition, we propose a new model named **PGCR** (Policy Guided Conversational Recommendation) for the new dataset. It mainly consists of three modules : Recommendation System, Policy Network and Response Generator. The Recommendation System and Response Generator roughly adopt the framework used in prior works (Zhou et al., 2020a). Yet, with the attendance of DAs, we are able to better formulate user profile by the immediate feedback for recommendation system. Also, through concatenating the utterance and its corresponding dialogue act and viewing the DAs as special tokens, the response generator can generate utterance more related with its act guided by the learned policy. Lastly, to show the advance of the DAs scheme, we design a two-stage training method for PGCR, i.e. supervised training and self-play (Silver et al., 2017; Vinyals et al., 2019; OpenAI, 2018). Th latter one is designed to maximize the success rate of recommendation and user's engagement of conversation.

We summarize our contributions as follows:

(1) Based on the Redial dataset, we formulate a new dataset DA-ReDial, which provide a relatively novel paradigm for open-ended CRS and shows great potential in this field.

(2) We propose a three-module model — PGCR, which can apprehend user's feedback from the perspective of DAs, maintain a better user profile and generate policy-guided response.

(3) Extensive experiments show that PGCR outperforms most baselines in recommendation and generation. In addition, the Policy Network fine-tuned by the devised self-play algorithm verifies the introduction of DAs can better control the dialogue policy, which facilitates the model interpretability.

2 Related Work

In this section, the application of Dialogue Acts (DAs) in Conversational Recommendation System (CRS) will be studied. Also, we will discuss CRS from a policy-guided perspective.

Dialogue Acts. Dialogue Acts (DAs), designed for utterance in dialogue, usually model dialogue structure and guide response generation explicitly or implicitly. In prior works (Sun and Zhang, 2018; Lei et al., 2020c,b; Deng et al., 2021), the response from the Generation Module is usually designed with pre-defined slots in advance. When receiving policy guidance, these slots will be filled in item-related words to recommend, query, or chat. The policy-guided templates, to some extent, can be regarded as implicit dialogue acts. For instance, the estimation component in the work (Lei et al., 2020b) can guide the system to choose an attribute to ask, or make a recommendation by user profile. However, these methods are not suitable for open-ended field, because the dialogue module needs to have the ability to generate nature response spontaneously. The work in (Liang et al., 2021) learns the response template in the way of Sequence-to-Sequence(seq2seq), making the task of generation easier. Yet, it is still unable control the sentence generation from a sentence-level due to lack of explicit dialogue act. Recently, Ma (Ma et al., 2021) uses a unique tree-structured reasoning on a Knowledge Graph (KG) to select entities as part of the dialogue act, and generate the response guided by the act. It also abstracts three kinds of generation policy — i,e, recommending, asking and chi-chat. However, its dialogue acts rely on complex tree structure, thus lacking generalization; in addition, it cannot understand seeker's intention from the perspective of dialogue acts in the conversation. Therefore, motivated by prior works (Traum, 1999; Takanobu et al., 2019, 2020; Ma et al., 2021) we exclusively designed a Dialogue Acts schema for the open-ended dataset ReDial (Li et al., 2018). The introduction of DAs in dataset enables open-ended CRS conveniently guide generation with explicit policy.

Policy-guided CRS. Policy-guided CRS tend to design policy for next utterance given dialogue history context. Zhou (Zhou et al., 2020b) formulates a topic-guided dataset and propose a model which could direct dialogue towards recommendation scenario. Yet, topic-guided strategy narrowly focus on

Speaker	DAs	Utterances
HUMAN:	CHAT	Hello!
SEEKER:	HINT BY QUESTION	Hello! I am looking for a comedy. Do you have any suggestions?
HUMAN:	RECOMMEND BY STATEMENT	Oh i love comedies, and i would suggest @97007. It is hilarious.
SEEKER:	ACCEPT	That's one of my favorites ! it is so funny , and also very suitable with parents .
HUMAN:	RECOMMEND BY QUESTION	Would you like to enjoy @126619?
SEEKER:	NEUTRAL	I have not seen that one. Is it just as good as the first one?

Table 2: Samples from DA-Redial. In data pre-processing, the DA label will be concatenated with the utterance and act as the first token to be decoded.

entity-level, which also cannot optimize the policy module like other works (Sun and Zhang, 2018; Kang et al., 2019; Lei et al., 2020c,b; Deng et al., 2021). The latter works focus on Policy Module and optimize the Policy Network to pursue a long-term reward through reinforce learning. In addition, inspired by the use of bot-play algorithm (Silver et al., 2017; Vinyals et al., 2019; OpenAI, 2018; Kang et al., 2019; Takanobu et al., 2020) , we design a self-play training between bot and seeker to optimize dialogue policy strategy and facilitate the interaction with seeker. Self-play algorithm demonstrates the control over dialogue generation through designed reward function. For Recommender Module, it essentially aims to formulate user profile and then recommend items based on user's preference. Knowledge Graph enable some works (Chen et al., 2019; Zhou et al., 2020a) to utilize external knowledge and model user's profile. However, these methods ignore the user's true intention since they cannot recognize the seeker's dialogue acts. Lei (Lei et al., 2020c) tries to narrow search space of user's preferred attributes through explicit policy strategy. Therefore, we also maintain a DA-aware user profile based on the policy incorporated with DAs and thus offer more accurate recommendation.

3 Dataset Construction

Dialogue Acts Inspire by prior works (Traum, 1999; Takanobu et al., 2019, 2020; Ma et al., 2021) in dialogue acts, we design 10 dialogue acts shown in table 1, which summarizes the most representative acts in Conversational Recommender System (CRS) dataset. For open-ended CRS, the bot agent aims to collection information and recommend items. Also, it is required to have the function of chit-chat. Thus, the five acts for bot can be representative. Seekers usually start the conversation with explicit goal of asking for recommendation. Besides the "CHAT" act, we also design five acts for seeker. We adopt a semi-automatic annotation method. Firstly, workers are employed to label the open dataset ReDial (Li et al., 2018). The dataset contains 10006 dialogues consisting of 182150 utterances, in which 1000 dialogues (including 8802 utterances) are labeled. Then, we train a DAs classifier based on the human-labeled data and weekly label the remaining dialogues in ReDial.

DAs Classifier We adopt a classifier of neural network based on BERT (Devlin et al., 2019), which includes a encoder to represent the text and full-connoted layer to predict the probability distribution of DAs. The labeled data has been split into train and valid set with a ratio of 8.5:1.5. In addition, we note that the distribution of DAs is uneven: some labels like 0,4 and 6 prevail while the other labels like 1 and 8 are sparse. Thus, we use upsampling method and some data augmentation approaches (Karimi et al., 2021; Gao et al., 2021). Table 2 shows a sampled case from the new dataset, and Figure 1 shows the distribution of DAs is basically consistent between human-labeled data and auto-labeled data. The accuracy ratio on valid set is 0.86, and the evaluation detail of other metrics can be seen in Appendix A. As shown in Figure 1, the distribution of DAs is basically consistent between human-labeled data and auto-labeled data. The dataset and implementation detail of code will be available on Github soon.

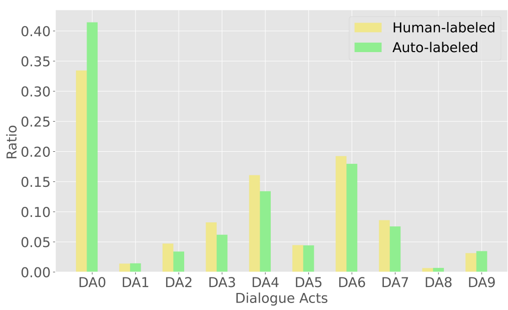

Figure 1: The distributions of DAs in human-labeled data and auto-labeled data.

4 Our Approach

In this section, we formally define the problem of Conversational Recommender System (CRS) in Section 4.1. Then, we detail the three different but connected modules of our proposed model PGCR (depicted in Figure 2) in Section 4.2-4.4. Finally, the training objective of PGCR is discussed in Section 4.3.

4.1 Problem Definition

Let m denotes a item from item set M, w represents a word in the vocabulary W, and e denotes entity from entity list E, which includes items and non-items. A dialogue D consists of a list of utterances $D = \{x_1, y_1, \cdots, x_{t-1}, y_{t-1}, x_t\}$, where x is utterance from user u, y is bot's utterance and t denotes the dialogue turn. Compared with traditional CRS, we also define a_t as the dailogue act for each utterence u_t. Further, the problem can be decomposed into three sub-tasks:

Item Recommendation Given the dialogue history D, the recommender system need to model user profile p_u firstly and then predict item m with high ranking.

DAs Prediction The input of this task is the dialoge history D. Then a Policy Network is used to predict the dialogue act a_t for next utterence u_t.

Response generation Given the item recommended m, the dialogue act a_t and the dialogue history D, the Response Generator is required to generate an utterance y_t guided by the dialogue act. In addition, the information of item m should be integrated in the utterance if necessary.

4.2 Recommend Module

KG-based Recommender As shown in Figure 2, our approach adopt a standard Knowledge Graph (KG)-based model following prior work (Zhou et al., 2020a). Firstly, the encoder of the reocommender module incorporate both word-oriented KG (Speer et al., 2017) and item-oriented KG (Bizer et al., 2009) to represent the use profile p_u. Specifically, entities $T = (e_1, e_2, ...e_N)$ are extracted from dialog history where entity e_i can be either item m or non-item w, and generate user's representation p_u:

$$p_u = \beta h_w + (1 - \beta) h_m \tag{1}$$

where h_w is non-item embedding, h_m is item embedding and β is the output of a gating network (Zhou et al., 2020a). With the representation, we are able to compute the score that recommend an item m to user through softmax function:

$$P_{\text{rec}}(m) = \text{softmax}\left(p_u^{\mathrm{T}} \cdot H_m\right) \tag{2}$$

where H_m is the hidden representation of item m learned from Knowledge Graph. Through the score function, we can rank all the items and make recommendation.

DA-aware user profile Prior works ususally construct user profile based on the entity list T, which includes all entities mentioned in context and linked to the KG. Yet, the method cannot fully utilize user's feedback to previous queries and recommendations. For instance, when a item is rejected, the entities related with the item might be a distribution to Knowledge Graph. It is not necessary infer user profile with that negative samples. In our dataset, the rejection feedback is an explicit act and is crucial to modeling the user profile. Inspired by the work in (Lei et al., 2020c), we adopt a simple method of reflection. When item rejected by user, we take the entities related with rejection as noise and delete them from the item list. Thus, we make an more interactive entity list, which help the module maintain a DA-aware user profile and offer high-quality recommendation.

4.3 Policy Network

Given the dialogue history $D = \{x_1, y_1, \cdots, x_t\}$, the policy network aims to predict a dialogue act a_t for next utterance y_t. Firstly, a BERT-based encoder (Devlin et al., 2019) is used to encode the dialogue history to get its hidden vector denoted by the [CLS] token (here we mainly focus on the last two utterances). A utterance-level LSTM (Hochreiter and Schmidhuber, 1997) is also used to generate hidden representation of context h_t:

$$h_t = \text{LSTM}(\text{BERT}(y_{t-1}), \text{BERT}(x_t)) \tag{3}$$

Last, a fully connected feed-forward network is used to compute the probability distribution of Dialogue Acts a_t:

$$\pi(a_t | h_t) = \text{softmax}\left(\text{FFN}(h_t)\right) \tag{4}$$

4.4 Response Generator

Transformer-based Generator The generator module is the pivot of the system, aiming to generate response with the information of item. We

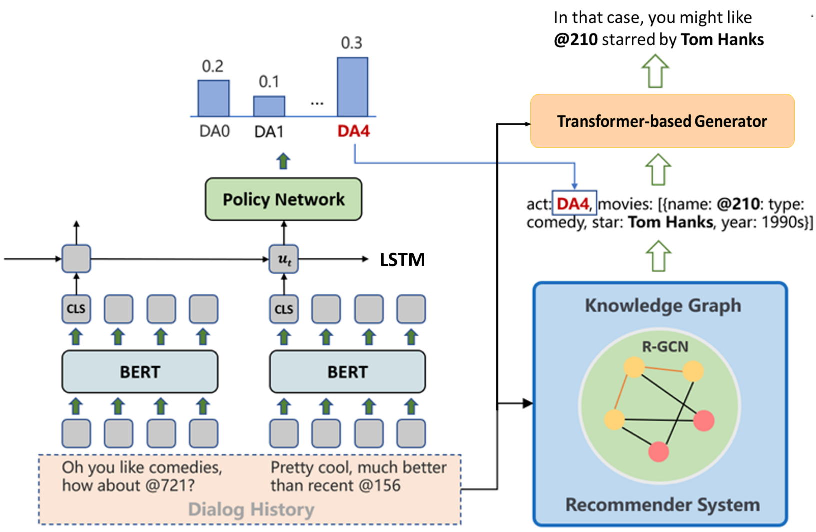

Figure 2: **The overview of our approach**. The Recommender System formulates user profile and predicts an item for user; the policy network generates the distribution of Dialog Acts. Pivoted on the two modules, Response Generator is able to generate DA-guided response with information of item.

use standard Transformer encoder architecture and the KG-based decoder (Zhou et al., 2020a), which can generate informative keywords or entities in response. The context D is fed into the Generator to get the KG-enhanced representation H_D. Then, at each decoding step, the decoder can generate a regular word from the vocabulary or a entity related with the recommended item.

$$P_{\text{gen}}(w_i|w_1,...w_{i-1}) = \text{softmax}(f(H_D)) \quad (5)$$

where $w_i \in W$ denotes word token and $f(\cdot)$ is the KG-based decoder. Thus, we get the probability of each output token.

DA-Guided Response To better guide the response generation with the designed DA schema, we take the 10 dialogue acts as special tokens and join them into vocabulary. Then, each utterance is concatenate with its DA label:

"[RECOMMEND_BY_QUESTION] Do you like Mission Impossible starred by Tom Hanks ?"

When inferering with the way of seq2seq, the DA label plays the role of the token $[BOS]$ of decoder. For isntance, when the label [RECOMMEDN_BY_QUESTION] is predicted firstly, an utterance can be subsequently generated step by step. Due to the auto-regression decoding mechanism, the act of the utterence might highly relate with the label. Thus, we are able to generate DA-guided response.

4.5 Training Objectives

We break down the holistic traning process into two stages, i.e. Supervised Training and Self-Play Tranning.

Stage I: Supervised Training In this stage, PGCR plays a good learner with all three modules trained supervised. Firstly, we optimize the recommender system with a standard cross-entropy loss:

$$L_{\text{rec}} = -\sum \log(P_{rec}(m)) \quad (6)$$

When the loss of the recommendation system converges, the policy network and generation module are jointly trained then. The loss function for this two modules are as follows:

$$L_{\text{da}} = -\sum \log(\pi(a_t|h_t)) \quad (7)$$

$$L_{\text{gen}} = -\sum_{i=1}^{N} \log\left(P_{\text{gen}}(w_i|w_1,...w_{i-1})\right) \quad (8)$$

Thus, the two modules perform gradient descent to update parameters by the loss: $L = L_{\text{da}} + L_{\text{gen}}$.

Stage II: Self-Play In this stage, we fix the Recommerder and Generator, and optimize Policy Network by Reinforce Learning maximize the rate of successful recommendation and increase user's engagement of conversations. This stage is a autogenic process between two agents—i.e. **BOT** and **SEEKER**.

Algorithm 1 The procedure of self-play between Bot and Seeker

Start with the first turn or two turns of real conversation $D_t = \{x_1, y_1, x_2\}$, turn $t = 2$
while True **do**
  Prepare context D_t for the bot
  Predict dialogue act a_t from the policy network $\pi(a_t|C_t)$
  Generate response y_t from the bot, guided by the a_t
  Prepare context for the seeker
  Generate response x_{t+1} from the seeker
  $t = t + 1$
  if Seeker quits *or* Beyond maximum number of iteration **then**
    $break$
  end if
end while
Calculate reward based on the the dialogue
Update the Policy Network by policy gradient.

Let PGCR act as the **BOT** to give response; we also create a stimulated **SEEKER** following the response generator of PGCR. As shown in Algorithm 1, the two agents are required to interact with each other. We compute reward (Lei et al., 2020b) based on the conversation generated in this episode. We design six kinds of rewards, i.e. (1) r_{acc}, a positive reward when seeker accpets the recommendation. (2) r_{neu} a weekly positive reward when seeker is neutral to the the recommendation. (3) r_{rej} a negative reward when seeker rejects the recommendation. (4)r_{quit} a negative reward when the seeker quits. (5) r_{sim} a negative reward to prevent strategy loop. (6) r_{int} a positive/negative reward on each turn to increase/decrease dialog interaction. In one episode, r_t denotes the immediate reward at t-turn, and R_t denotes the total reward accumulating from turn t to the final turn T: $R_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$, where γ is a discount factor. When it is bot's turn, the policy network π returns the probability of dialogue acts. We update the parameters by policy gradient algorithm:

$$\theta \leftarrow \theta - \alpha \frac{d}{d\theta} log(\pi_\theta(a_t|s_t))R_t \qquad (9)$$

Reward	Strategy I	Strategy II
r_{int}	-0.2	$+0.1$
Decaying r_{acc}	✓	✗

Table 3: Strategy I focuses on recommendation task and aims to let seeker accept the recommendation as soon as possible; Strategy II focuses on the interactions with seeker and enhances the engagement of seeker in conversation. Specifically, the decaying $r_{acc} = \beta^t$, where $\beta = 0.9$ is the decaying factor and t denotes the t-turn. Both of them share following rewards: $r_{neu} = 0.2, r_{quit} = -0.2, r_{sim} = -0.5$, discount facot$\gamma = 0.95$.

5 Experiment

5.1 Setup

Dataset We evaluate our approach on DA-ReDial introduced in Section 3. At the first stage — Supervised Traning, we split DA-ReDial dataset into training, validation, and test set in an 80-10-10 proportion. At the self-play training, we randomly choose the first one or two turns of dialog from the DA-ReDial dataset.

Implementation details The models are implemented in Pytorch and trained on an NVIDIA 3090. We use bert-base-cased as the encoder of Policy Network. The main hyperparameter settings of Recommender System and Resonse Gnererator follow the work of Zhou (Zhou et al., 2020a). The embedding dimension of Generator is set to 300, while the embedding dimension of Recommber is 128. We train the model with 64 batch size, Adam optimizer and 0.001 learning rate. To show the control of Dialogue Act over dialog, we design two sets of reward function to fine-tune the Policy Network in Table 3.

Evaluation Metrics For recommendation task, the evaluation consists of Recall@k (k=1, 10, 50) following works (Li et al., 2018; Chen et al., 2019; Zhou et al., 2020a), item ratio and item diversity. Recall@K measures whether the top-k predicted items contain the groud-truth; item diversity and item ratio measure the quantity and diversity of items incorporated into the response. For conversation task, we adopts perplexity(ppl) to measures the fluency of the generated response. Also, Distinct n-gram (n = 2, 3, 4) (Li et al., 2016) are used to measures the diversity of response at a sentence-level, which are related with the number of distinct n-grams. For Policy Network, we show the distribution of DAs predicted for impending utterance.

Model	PPL	Dist-2	Dist-3	Dist-4	R@1	R@10	R@50	Item Diversity	Item Ratio
REDIAL (Li et al., 2018)	28.1	0.225	0.236	0.228	0.024	0.140	0.320	-	15.8
KBRD (Chen et al., 2019)	17.9	0.263	0.368	0.423	0.031	0.150	0.336	-	29.6
KGSF (Zhou et al., 2020a)	5.55	0.305	0.466	0.589	0.039	0.183	0.378	6.03	31.5
CR-Walker	-	-	-	-	0.040	0.187	0.376	-	-
RID	54.1	0.518	0.624	0.598	-	-	-	-	43.5
NTRD	**4.41**	0.578	0.820	1.005	-	-	-	**11.05**	66.77
PGCR	8.71	**0.631**	**1.142**	**1.493**	**0.042**	**0.207**	**0.406**	9.24	**80.1**

Table 4: Automatic evaluation results on the DA-REDIAL dataset. Numbers in bold denote the best performance.

Further, with two different strategies (Table 3), we show how different training objectives affect the strategy of Dialogue Acts and the metrics we care about.

5.2 Baselines

We introduce the baselines for the experiment in the following:

REDIAL (Li et al., 2018) offers a benchmark dataset Redial and adopt a generation module based on HRED (Serban et al., 2017).

KBRD (Chen et al., 2019) propose a KG-enhanced recommender to improve user representation and generate response with hgih-quality recommendations.

KGSF (Zhou et al., 2020a) incorporate external knowledge through a word-oriented KG and an item-oriented KG to enhance the Recommender Module and Generation Module.

CR-Walker (Ma et al., 2021) takes advantage of tree structured reasoning on KG and response with dialog acts guided.

NTRD (Liang et al., 2021) learns a neural template and insert item information into the pre-set slots.

RID (Wang et al., 2021) improves the performacne of CRS with pre-trained language model and knowledge graph.

Method	Item Diversity	Iitem Ratio
PGCR w/o s-p	9.24	80.1
PGCR w/ s-p1	**10.12**	**84.2**
PGCR w/ s-p2	7.93	66.2

Table 5: The comparison of item evaluation between PGCR without self-play (s-p) and PGCR with self-play (including strategy 1 and 2).

5.3 Main Results

Reommendation Table 4 shows the comparation of the evaluation of the baseline models and

Model	PPL	Dist-2	Dist-3	Dist-4
PGCR w/o s-p	8.71	0.631	1.142	1.493
PGCR w/ s-p1	**8.67**	0.603	1.020	1.334
PGCR w/ s-p2	9.03	**0.781**	**1.205**	**1.639**

Table 6: The comparison of generation between PGCR without self-play (s-p) and PGCR with self-play (including strategy 1 and 2).

our proposed method in item recommendation. In terms of Recall@k, KBRD and KGSF perform better than ReDial with external information from knowledge graph. CR-Walker outperms KGSF on Recall@1 and Recall@10 via its unique treestructe reaning graph. Founded on DA-aware user profile, our model outperforms all baseline models on R@k, which indicate the the introduction of DAs perfects the modeling of user profile and facilitate the recommener system.

In addition, NTRD, due to its novel item selector enable it generate more diversified items. Guided by DA label, our model performs best in item ratio, which means that the items can be better incoporated into response. We also note that when PGCR is fine-tuned by Srategy I, it performs better on item metrics.

Item diversity drops a little compared with NTRD (11.05 vs.9.25) though, our model still outperforms all baselines on item ratio by a large margin, which means the model can incorporate more recommended items into the system.

Generation From Table 4, PGCR outperforms all baselines on language diversity (disk-k). Comparing the DAs distribution between generated response (Figure 3) and the original dataset (The bot part of Figure 1), we conclude the introduction of DAs help the system simulate true distribution of dialogue. Thus, our model prevails in diversity of generation. NTRD maintains the best performance in Perplextity since the learning of response templates makes generation task easier.

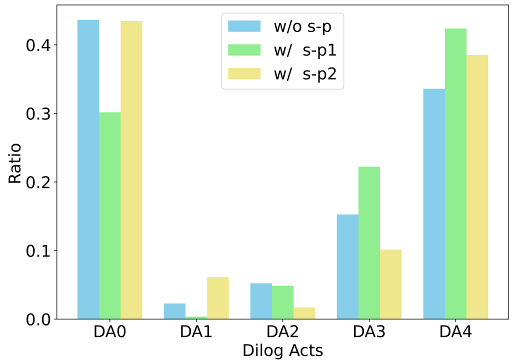

Figure 3: The distributions of DAs in response generated from different strategies.

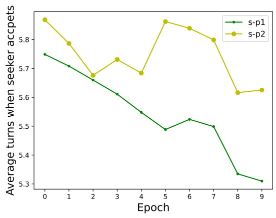

Figure 4: Average turns when items accepted

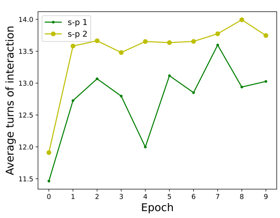

Figure 5: Maximum turns of dialogue

From table 6, we note that although the generation model is not directly optimized, the result is that strategy II not only maintains the fluency of the model (a little drop in PPL), but also improves the generation diversity.

Policy Network We report the comparison of PGCR without self-play and PGCR with different strategies of self-play on item evaluation and generation in Table 5 and 6. With the recommendation-oriented goal, the strategy I has an advantage on both item ratio and diversity, which indicates the response Generator can incorporate more items in diversity and quantity (Table 5). The same metrics drops in strategy II, since the interaction-oriented goal undermine the distribution of DAs, which is shown in Figure 3. Yet the strategy helps improve the performance of generation (Table 6), although it do not directly optimize the response generator but adapt the DAs distribution to the module.

Further, we draw the change of average turns during training process in Figure 4 and Figure 5. As we expected, with strategy I, the average turn when recommendation is firstly accepted basically drops with a little disturbance; yet, the average turn fluctuates a lot with strategy II although it could extend the maximum number of interactions. This further proves that Policy Network fine-tuned by self-play algorithm can explicitly adapt the model to the pre-defined goal.

Conversation Case	
SEEKER:	[HINT BY QUESTION] Hey! Can you suggest a movie to watch for date night please?
BOT:	[WHAT QUESTION] Hello, what kind of movie do you like?
SEEKER:	[HINT BY QUESTION] Have you seen any good movies ?
BOT:	[CHAT] I have seen the new @83552.
SEEKER:	[HINT BY STATEMENT] I have seen that one. I really liked it.
BOT:	[RECOMMEND BY STATEMENT] Did you see any of the new @187028?
SEEKER:	[NEUTRAL] I have not seen that one . What is it about ?
BOT:	[CHAT] It seem to be a love story.
SEEKER:	[ACCEPT] Oh, really? I want see it.

Table 7: A sampled case between the stimulated seeker and the bot during the satage of self-play.

Table 7 shows a sampled case between the stimulated seeker and the bot. It is clear that out model could generate policy-guided response and each utterance is highly correlated with the first [DA] token, which greatly gains more control and interpretability over dialogue.

6 Conclusion

In this work, we introduce a new dataset DA-ReDial designed by a schema with dialogue acts, which demonstrates a novel paradigm for conversational recommendation system. Based on DA-ReDial, a framework, including three modules – Recommender, Generator, Policy Network, is presented. Apart from the significant improvement over the baseline models, the framework shows outstanding explainability and controllability for CRS. Moreover, our experiment also indicates Dialogue Acts can explicitly mode the discourse structure of conversation and thus better guide the response generation.

We believe that the new dataset DA-ReDial, especially the part of dialogue acts, may provide a new paradigm for open-ended CRS. Our work tap the potential for future directions including (1) design better dataset with better DAs schema in open-ended CRS; (2) optimize the Policy Network with more reasonable strategy to stimulate real situation.

References

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. Dbpedia-a crystallization point for the web of data. *Journal of web semantics*, 7(3):154–165.

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.

Zhongxia Chen, Xiting Wang, Xing Xie, Mehul Parsana, Akshay Soni, Xiang Ao, and Enhong Chen. 2020. Towards explainable conversational recommendation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 2994–3000. ijcai.org.

Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H. Chi. 2018. Q&r: A two-stage approach toward interactive recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 139–148. ACM.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv e-prints*, page arXiv:2104.08821.

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.

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y-Lan Boureau, and Jason Weston. 2019. Recommendation as a communication game: Self-supervised bot-play for goal-oriented dialogue. 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 1951–1961, Hong Kong, China. Association for Computational Linguistics.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. Aeda: An easier data augmentation technique for text classification.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

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. *Proceedings of the 13th International Conference on Web Search and Data Mining*.

Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020b. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. *Proceedings of the 13th International Conference on Web Search and Data Mining*.

Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020c. 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. 2016. 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.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *NIPS*, pages 9725–9735.

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.

Lizi Liao, Ryuichi Takanobu, Yunshan Ma, Xun Yang, Minlie Huang, and Tat-Seng Chua. 2019. Deep conversational recommender in travel. *arXiv preprint arXiv:1907.00710*.

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, and Minlie Huang. 2021. Cr-walker: Tree-structured graph reasoning and dialog acts for conversational recommendation.

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

OpenAI. 2018. Openai five. `https://blog.openai.com/openai-five/`.

Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE.

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.

Heung-Yeung Shum, Xiaodong He, and Di Li. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. *arXiv preprint arXiv:1801.01957*.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature*, 550(7676):354.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *SIGIR*, pages 235–244.

Ryuichi Takanobu, Runze Liang, and Minlie Huang. 2020. Multi-agent task-oriented dialog policy learning with role-aware reward decomposition.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog.

David Traum. 1999. Speech acts for dialogue agents.

Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M. Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Yuhuai Wu, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. 2019. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/.

Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Kam-Fai Wong, and Daxin Jiang. 2021. Finetuning large-scale pre-trained language models for conversational recommendation with knowledge graph.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018a. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Tong Zhang, Yong Liu, Peixiang Zhong, Chen Zhang, Hao Wang, and Chunyan Miao. 2021. Kecrs: Towards knowledge-enriched conversational recommendation system. *arXiv preprint arXiv:2105.08261*.

Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018b. 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, 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.

A Appendix

Here, we report the evaluation results of classification introduced in Section 3.

DAs	Precision	Recall	F1
0	0.84	0.86	0.85
1	0.83	0.8	0.82
2	0.91	0.87	0.89
3	0.92	0.92	0.92
4	0.92	0.87	0.80
5	0.83	0.85	0.84
6	0.84	0.86	0.85
7	0.77	0.81	0.79
8	0.71	0.83	0.77
9	0.84	0.75	0.80

Table 8: Automatic evaluation results of classification on valid set.