

CR-Walker: Tree-Structured Graph Reasoning and Dialog Acts for Conversational Recommendation

Wenchang Ma^{1*}, Ryuichi Takanobu^{2*}, Minlie Huang^{1†}

¹CoAI Group, DCST, IAI, BNRIST, Tsinghua University, Beijing, China

²Alibaba Group, Hangzhou, China

mwc17@mails.tsinghua.edu.cn, ryuichi.gxly@alibaba-inc.com, aihuang@tsinghua.edu.cn

Abstract

Growing interests have been attracted in Conversational Recommender Systems (CRS), which explore user preference through conversational interactions in order to make appropriate recommendation. However, there is still a lack of ability in existing CRS to (1) traverse multiple reasoning paths over background knowledge to introduce relevant items and attributes, and (2) arrange selected entities appropriately under current system intents to control response generation. To address these issues, we propose CR-Walker in this paper, a model that performs tree-structured reasoning on a knowledge graph, and generates informative dialog acts to guide language generation. The unique scheme of tree-structured reasoning views the traversed entity at each hop as part of dialog acts to facilitate language generation, which links how entities are selected and expressed. Automatic and human evaluations show that CR-Walker can arrive at more accurate recommendation, and generate more informative and engaging responses.

1 Introduction

Many researches have been drawn to combine conversational assistants with recommender agents into one framework due to its significance and value in practical use (Sun and Zhang, 2018; Jan-nach et al., 2020), but creating a conversational recommender system (CRS) that bridges conversation and recommendation still remains challenging.

One of the challenges lies in **reasoning over the background knowledge for accurate recommendation**. Prior studies usually focused on using context and knowledge as an enrichment to recommendation (Chen et al., 2019; Zhou et al., 2020a), but neglected to fully exploit the connec-

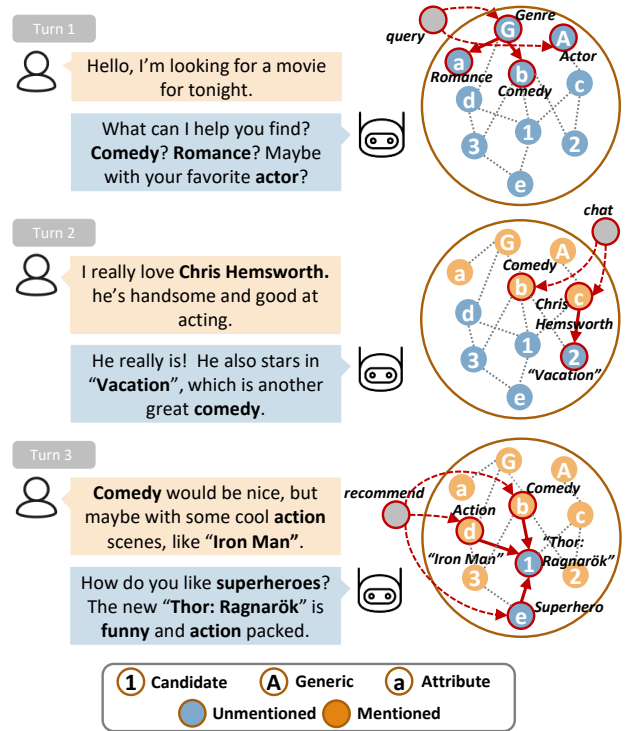


Figure 1: First three turns of an example dialog. The dialog is shown on the left with entities on the KG in bold. The graph on each dialog turn's right demonstrates the reasoning process of CR-Walker, with the reasoning tree marked red. Throughout this paper, candidate items are noted with numbers, and generic classes / attributes with upper-/lower-case letters. The orange/blue color indicates that the entity is mentioned/unmentioned in the previous context.

tion between entities¹ to infer the system action. In particular, this requires the system to make *multi-path reasoning* on background knowledge, since one entity may have multiple relationships with different entities through multi-hop reasoning. For example in Fig. 1, after the user mentioned "Hemsworth", the agent chats on "Vacation" starring "Hemsworth", and further explores the user

*Equal contribution.

†Corresponding author.

¹An entity can be any node on a knowledge graph throughout this paper, including items and their attributes. The definition is provided in Sec. 3.1.

interests in “Comedy” movies. It then recommends “Thor” based on several distinct paths of reasoning over user preference (“comedy” & “action”).

Another challenge lies in **fully utilizing the selected entities in response generation**. Since different dialog actions can be applied in conversational recommendation, selected entities need to be properly expressed with the guide of *dialog acts*, an abstract representation of dialog semantics and intentions, in order to form natural, informative, and engaging utterances to interact with users. However, most previous works (Moon et al., 2019; Lei et al., 2020a) stopped at inferring entities without modeling response generation. In Fig. 1 again, the agent first asks the user’s preferred genres and actors, and then talks about the star and the movie to engage the user in the conversation, and last recommends a movie based on the user interests. In addition, the agent provides explanations at the third turn to make recommendation more interpretable and persuasive.

To address these issues, we propose **Conversational Recommendation Walker (CR-Walker)** in this paper. It first selects a system intent to decide whether the system asks for information, chats about something, or makes a recommendation. Then, it performs tree-structured reasoning on a knowledge graph (KG) and dialog context, creating a reasoning tree comprised of relevant entities to be introduced in response. The hierarchical arrangement of entities on the tree preserves the logical selection order under current system intents, which is transformed to dialog acts. The linearized representation of dialog acts further guides on generating informative and engaging responses with pre-trained language models. Results show that CR-Walker outperforms strong CRS on two public datasets in recommendation and generation tasks.

In brief, our contributions are summarised below²: (1) CR-Walker conducts tree-structured reasoning on a knowledge graph and dialog context to explore background knowledge and exploit connection between entities for more accurate recommendation; (2) CR-Walker transforms the reasoning tree into dialog acts that abstract the semantics and hierarchy of selected entities, and thereby generates more engaging responses for recommendation; (3) We evaluate CR-Walker on two conversational recommendation datasets, achieving outstanding

performance in automatic and manual evaluation, from both recommendation and conversation aspects.

2 Related Work

Conversational Recommender Systems (CRS) learn and model user preference through dialog, which support a richer set of user interactions in recommendation (Jannach et al., 2020). Previous CRS can be roughly categorized into two types.

One is recommendation-biased CRS (Sun and Zhang, 2018; Zhang et al., 2018, 2020; Zou et al., 2020) that asks questions about user preference over pre-defined slots or attributes to recommend items. As system response can be grouped into some pre-defined intents, it can be implemented with the help of language templates. Under this simplified setting, approaches of this type do not model language generation explicitly (Lei et al., 2020a,b). Such dialogs can only provide limited actions without revealing why the system makes such recommendation (e.g. by asking on a fixed set of attributes) (Christakopoulou et al., 2016, 2018), thus leading to unsatisfactory user experience. Recently, Moon et al. (2019) improves knowledge selection by assuming a single chain of reasoning throughout the conversation. It relies on fine-grained annotations that follow single-path reasoning scheme. However, multiple entities can be selected at each reasoning hop (e.g. recommend several items within one turn, each item with different reasons). Therefore, we propose tree-structured reasoning in this work to enable CRS to select multiple entities through multi-path reasoning for accurate recommendation. Xu et al. (2020) introduces a dynamic user memory graph to address the reasoning of user knowledge in CRS, which is beyond the scope of this paper.

The other is dialog-biased CRS (Li et al., 2018; Kang et al., 2019; Liao et al., 2019; Liu et al., 2020) that makes recommendations using free text, which have much flexibility to influence how the dialog continues. As these systems suffer from existing limitations in NLP (e.g. understand preference implicitly from user expression), most methods incorporate external information such as KG and user logs to enhance the dialog semantics (Yu et al., 2019; Zhou et al., 2020a) or update the user representation (Zhang et al., 2019; Chen et al., 2019). However, these methods do not capture higher-level strategic behaviors in recommendation

²The codes are released at <https://github.com/truthless11/CR-Walker>

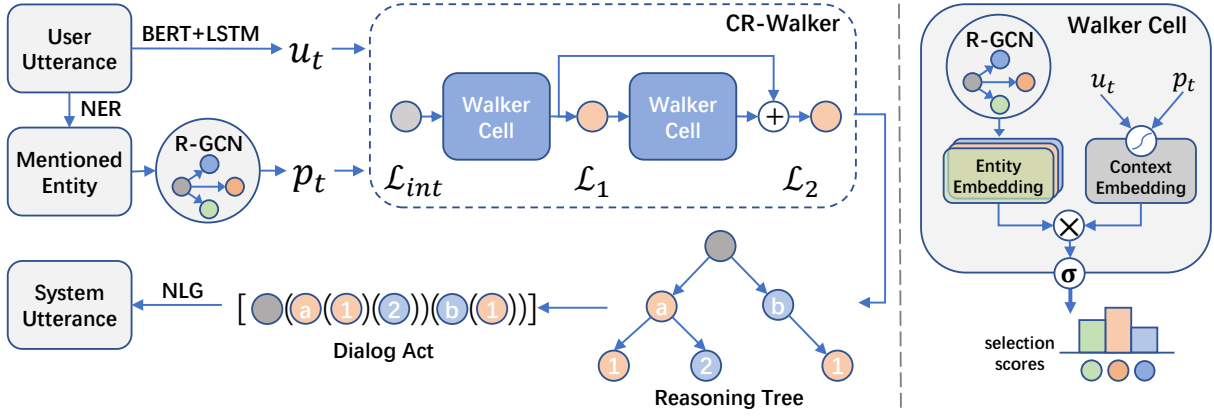


Figure 2: Left: Illustration of CR-Walker’s overall architecture. CR-Walker first decides the system intent and then applies walker cells to perform tree-structured reasoning on the knowledge graph in two stages. The transformed dialog acts are used to guide response generation. Right: Detailed structure for a single walker cell. A walker cell calculates the similarity between the entities on a graph and the context embedding that integrates utterance embedding and user portrait. Entity selection is learned by logistic regression to enable multiple selections.

to guide the conversation. To solve this issue, Zhou et al. (2020b) incorporates topic threads to enforce transitions actively towards final recommendation, but it models CRS as an open-ended chit-chat task, which does not fully utilize relations between items and their attributes in response. In contrast, CRS can be regarded as a variation of task-oriented dialog system that supports its users in achieving recommendation-related goals through multi-turn conversations (Tran et al., 2020). Inspired by the use of dialog acts (Traum, 1999), we choose a set of system dialog acts in CRS to facilitate information filtering and decision making as task-oriented dialog policy (Takanobu et al., 2019, 2020) does.

3 CR-Walker: Conversational Recommendation Walker

In this section, we start from defining the key concepts of knowledge graph and dialog acts used in CR-Walker. As illustrated in Fig. 2, CR-Walker works as follows: First of all, dialog history is represented in two views: one is utterance embedding in the content view, and the other is user portrait in the user interest view. Then, CR-Walker makes reasoning on a KG to obtain a reasoning tree, which is treated as a dialog act. Afterwards, the tree-structured dialog act is linearized to a sequence, on which CR-Walker finally generates responses with a conditional language generation module.

3.1 Key Concepts

We construct a knowledge graph $G = (E, R)$ as follows: the entities E on the graph are divided

into three categories, namely **candidate items**, **attributes**, and **generic classes**. There are various relations R among these entities. Each candidate item is related to a set of attributes, while each attribute is connected to its corresponding generic class. There might also exist relationships between different attributes. Taking movie recommendation as an example, the candidate movie *Titanic* is linked to attributes *Romance*, *Leonardo DiCaprio* and *James Cameron*, and these three attributes are linked to generic classes *Genre*, *Actor* and *Director*, respectively.

We also define a set of system actions in CRS. We abstract three different system intents to represent actions commonly used in a dialog policy: **recommendation** that provides item recommendation and persuades the user with supporting evidence, **query** that asks for information to clarify user needs or explore user preference, and **chat** that talks on what has been mentioned to drive the dialog naturally and smoothly. Example utterances of three intents are shown in Fig. 1. Then, we define a dialog act A as an assembly of a system intent and entities selected by the system, along with their hierarchy relations.

3.2 Reasoning Process

CR-Walker learns to reason over KG to select relevant and informative entities for accurate recommendation and generating engaging conversations. Considering the large scale of KG and different system actions in CRS, we design several two-hop reasoning rules to help CR-Walker narrow down the search space, thereby making the reasoning pro-

Intent	Hop 1	Hop 2
Recommend	mentioned attributes + mentioned items’ attributes	candidate items
Query	generic classes	attributes
Chat	mentioned entities	all entities

Table 1: Reasoning rules for narrowing down CR-Walker’s search space. For each system intent, we only maintain the legal entities at each hop during reasoning.

cess more efficient on large KG. As shown in Table 1, all the reasoning rules are designed in line with the conceptual definition of corresponding intents. The reasoning process of CR-Walker starts from one of the three intents. It then tries to explore intermediate entities as a bridge to the final recommendation, and finally reaches the target entities at the second hop.

As explained in Sec. 2, multiple entities can be selected at each hop in CRS, therefore forming a tree structure on the graph instead of a single path as in previous work (Moon et al., 2019). The child entities at the second hop are *neighboring* to their parent entities at the first hop on the graph, except when the intent is “recommend”. We allow all candidate items to be recommended, even if some of them have no connection with other entities on the graph. In addition, we maintain the status of each entity whether the entity is mentioned or not in the context, to facilitate reasoning during interaction.

3.3 Dialog and Knowledge Representation

In this subsection, we describe how to represent dialog context, external knowledge and user interests in CR-Walker.

Utterance Embedding We formulate the dialog history $D = \{x_1, y_1, \dots, x_{t-1}, y_{t-1}, x_t\}$, where x_t and y_t is user/system utterance respectively. At each dialog turn t , we first use BERT (Devlin et al., 2019) to encode last system utterance y_{t-1} and current user utterance x_t successively. The embedding of “[CLS]” token of x_t is applied as the turn’s representation, denoted as $\text{BERT}([y_{t-1}; x_t])$. Then the utterance embedding \mathbf{u}_t is obtained simply through a LSTM over $\text{BERT}([y_{t-1}; x_t])$ to capture the sentence-level dependencies. Formally,

$$\mathbf{u}_t = \text{LSTM}(\mathbf{u}_{t-1}, \text{BERT}([y_{t-1}; x_t])). \quad (1)$$

The hidden state of LSTM $\mathbf{u}_t \in \mathbb{R}^d$ is taken as the utterance embedding to represent dialog context.

Entity Embedding To introduce external structured knowledge in CR-Walker, we extract KG from DBpedia (Auer et al., 2007) and add generic classes (see Sec. 3.1). We encode the graph using R-GCN (Schlichtkrull et al., 2018), by virtue of its capability of modeling neighboring connections more accurately by considering different relations. Formally, for each entity $e \in E$, the entity embedding $\mathbf{h}_e^{(l)} \in \mathbb{R}^d$ at each layer l is calculated as:

$$\mathbf{h}_e^{(l+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{e' \in \mathcal{N}_e^r} \frac{1}{jN_e^r} \mathbf{W}_r^{(l)} \mathbf{h}_{e'}^{(l)} + \mathbf{W}_0^{(l)} \mathbf{h}_e^{(l)}\right), \quad (2)$$

where \mathcal{N}_e^r denotes the set of neighboring entities of e under the relation r , and $\mathbf{W}_r^{(l)}, \mathbf{W}_0^{(l)} \in \mathbb{R}^{d \times d}$ are learnable matrices for integrating relation-specific information from neighbors and the current layer’s features respectively. At the final layer L , the embedding $\mathbf{h}_e^{(L)}$ is taken as the entity representation, and is denoted as $\mathbf{h}_e \in \mathbb{R}^d$ in the following text.

User Portrait We build a user portrait to represent user interests using both dialog and KG here. Given the dialog history, we performed *named entity recognition* (NER) to identify entities mentioned in the previous user utterances $\{x_1, \dots, x_{t-1}, x_t\}$ using spaCy, then linked them to the entities in the KG with simple fuzzy string matching. The status of identified entities is updated as “mentioned”. We thus obtain all the representation of mentioned entities $\mathbf{M}_t \in \mathbb{R}^{d \times |\mathcal{M}_t|}$ by looking up entity embedding:

$$\mathbf{M}_t = (\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{|\mathcal{M}_t|}).$$

Following Chen et al. (2019), we calculate the user portrait $\mathbf{p}_t \in \mathbb{R}^d$ via self-attention:

$$\begin{aligned} \boldsymbol{\alpha}_t &= \text{softmax}(\mathbf{w}_p \tanh(\mathbf{W}_p \mathbf{M}_t)), \\ \mathbf{p}_t &= \boldsymbol{\alpha}_t \mathbf{M}_t. \end{aligned} \quad (3)$$

3.4 Tree-Structured Graph Reasoning

The reasoning process of CR-Walker initiates from a system intent as the start point on the graph, and expands into multiple paths to get a reasoning tree.

First of all, we treat intent selection as a simple 3-way classification problem parameterized by θ_i :

$$\begin{aligned} p_{\theta_i}(y_t^{int} | x_t) &= \text{softmax}(\mathbf{W}_{int}^2 \text{ReLU}(\mathbf{W}_{int}^1 \mathbf{u}_t)), \\ L_{int} &= -\log p_{\theta_i}(y_t^{int} | x_t). \end{aligned} \quad (4)$$

Noting that we only use utterance embedding \mathbf{u}_t as input, since we empirically find that introducing \mathbf{p}_t does not promote the performance of intent selection.

To expand a system intent into a reasoning tree, we propose the *walker cell*, a neural module shown in Fig. 2. Particularly, each time a walker cell C performs one-hop reasoning to select entities, to expand the tree from a given intent i , or a given entity e at hop $n=1$ or >1 respectively. It first integrates the dialog history representation via a gate mechanism to obtain context embedding \mathbf{c}_t :

$$\begin{aligned} \gamma^{(n)} &= \begin{cases} \sigma(\mathbf{w} [\mathbf{u}_t; \mathbf{p}_t; \mathbf{i}_t]), & n=1 \\ \sigma(\mathbf{w} [\mathbf{u}_t; \mathbf{p}_t; \mathbf{i}_t; \mathbf{h}_e]), & n>1 \end{cases} \\ \mathbf{c}_t^{(n)} &= \gamma^{(n)} \mathbf{u}_t + (1 - \gamma^{(n)}) \mathbf{p}_t, \end{aligned} \quad (5)$$

where $\mathbf{i}_t \in \mathbb{R}^d$ indicates trainable embedding of the selected intent. The cell then outputs the score of each entity e' using its entity embedding $\mathbf{h}_{e'}$:

$$\hat{s}_{e'} = \sigma(\mathbf{h}_{e'} \sum_{j=1}^n \mathbf{c}_t^{(j)}). \quad (6)$$

The estimated selection score $\hat{s}_{e'}$ indicates whether e' is selected for tree expansion. By incorporating $\mathbf{c}_t^{(j<n)}$, the current reasoning hop n is aware of the previous reasoning hop j . We describe this process of applying a single walker cell for entity selection from e (similar for intent i) as a function below:

$$\text{WALK}(e) = \mathbb{1}_{\hat{s}_{e'} > \tau, e' \in Z_e^{(n)}}, \quad (7)$$

where $Z_e^{(n)}$ is the set of legal entities to be selected starting from e according to the reasoning rule in Sec. 3.2, and τ is a threshold hyper-parameter.

In practice, we select at most m entities at hop 1 to control the reasoning tree's width. The reasoning path stops when no entities are selected at a certain hop or reaches hop 2.

3.5 Conditional Language Generation

Having selected the entities on the reasoning tree, we generate system response y_t conditioned on the user utterance x_t and tree-structured dialog act A_t . We formulate this as a language generation problem. The goal is to build a statistical model parameterized by θ_g as follows:

$$p_{\theta_g}(y_t | x_t, A_t) = \prod_{k=1}^K p_{\theta_g}(y_{k,t} | y_{<k,t}, x_t, A_t). \quad (8)$$

To facilitate response generation using a pre-trained language model (PLM), we convert the dialog act into a token sequence. As a dialog act of CR-Walker contains an intent and selected entities, and it is arranged in a tree structure, we can linearize the dialog act into a token sequence in the same way that a parser serializes a tree into a string with *preorder traversal*. As shown in Fig. 2, the brackets characterize the hierarchy of the dialog act with regard to the logical order of entity selection.

In this paper, we employ GPT-2 (Radford et al., 2019) as the backbone for response generation, where the model successively encodes the user utterance x_t and sequence dialog act A_t as input, and then decodes the response y_t in an auto-regressive generation process. During inference, Top- p sampling (Holtzman et al., 2020) is used for response decoding.

3.6 Model Optimization

At each turn t , we train the parameters of walker cells θ_w at each hop n using standard logistic regression loss:

$$\begin{aligned} L_n &= \sum_{e \in \mathcal{E}_t^{(n-1)}} \sum_{e' \in \mathcal{Z}_e^{(n)}} s_{e'} \log(\hat{s}_{e'}) \\ &\quad (1 - s_{e'}) \log(1 - \hat{s}_{e'}), \end{aligned} \quad (9)$$

where $s_{e'} \in \{0, 1\}$ is the label indicating the selection of entity e' , and $\mathcal{E}_t^{(n-1)}$ denotes the extracted entity set at dialog turn t at hop $n-1$ ³. Training the generation model is performed via maximizing the log-likelihood (MLE) of the conditional probabilities in Eq. 8 over the user utterance:

$$L_{gen} = \sum_{k=1}^K \log p_{\theta_g}(y_{k,t} | y_{<k,t}, x_t, A_t). \quad (10)$$

Noting that we use the extracted dialog acts in the corpus during training.

We jointly optimize all trainable parameters mentioned above. The final loss for optimization L is a weighted sum of all losses⁴:

$$L = L_{int} + \sum_{n=1}^2 \lambda_n L_n + L_{gen}. \quad (11)$$

³Specially, $\mathcal{E}_t^{(0)}$ is the singleton of selected intent.

⁴The outlined algorithm and implementation details of CR-Walker are presented in the appendix.

4 Experimental Setting

4.1 Data

We use two public conversational recommendation datasets to verify the effectiveness of CR-Walker. (1) ReDial (Li et al., 2018) is collected by crowdsourcing workers from Amazon Mechanical Turk (AMT). Two paired workers are assigned with a role of either recommender or seeker. At least 4 different movies are mentioned in each conversation. Each movie mentioned in the dialog is annotated explicitly. (2) GoRecDial (Kang et al., 2019) is collected in a similar way using ParlAI. In each dialog, each worker is given a set of 5 movies with corresponding descriptions. The seeker’s set represents his or her watching history, and the recommender’s set represents candidate movies to choose from. The recommender should recommend the **correct** movie among the candidates to the seeker. We then construct the KG and perform entity linking separately for GoRecDial and Redial.⁵

4.2 Baselines

We have compared CR-Walker with several strong approaches in Redial: (1) ReDial (Li et al., 2018): a benchmark model of Redial that applies an autoencoder recommender, a RNN-based NLG and a sentiment prediction module. (2) DCR (Liao et al., 2019): *Deep Conversational Recommender* uses a pointer network to incorporate global topic control and GCN-based recommendation in response generation. (3) KBRD (Chen et al., 2019): *Knowledge-Based Recommender Dialog* enriches user representation with a KG to give responses and recommendation following the user interests. (4) KGSF (Zhou et al., 2020a): *KG-Based Semantic Fusion* incorporates both word-oriented and entity-oriented KGs to enhance the data representations in CRS.

We also adopt several conversation recommendation methods as the baselines in GoRecDial: (1) BERT (Devlin et al., 2019): A BERT fine-tuned on GoRecDial, which encodes dialog contexts and movie descriptions. BERT features are used for response retrieval and movie recommendation. (2) R-GCN+GPT: A joint model combining a R-GCN (Schlichtkrull et al., 2018) for movie recommendation with a Transformer-based language model (Vaswani et al., 2017) for response generation. The movies are scored using similar structures within

⁵The KG construction details and dataset statistics are shown in the appendix.

our walker cell by calculating the dot-product between encoder hidden states and R-GCN embeddings. (3) GoRecDial (Kang et al., 2019): a benchmark model of GoRecDial, which is trained via multi-task supervised learning and bot-play learning by formulating the recommendation task as a task-oriented game.

5 Results

Model	Recommendation			
	R@1	R@10	R@50	Cov.
ReDial	2.3	12.9	28.7	5.8
DCR	2.7	14.8	30.6	1.5
KBRD	3.0	16.3	33.8	11.2
KGSF	3.9	18.3	37.8	12.2
CR-Walker	3.7	17.6	37.1	14.7
-depth=1	2.9	14.9	34.3	11.4
+ConceptNet	4.0	18.7	37.6	15.2

Table 2: Automatic evaluation of recommendation tasks on *ReDial*.

Model	Generation			Knowledge		
	BLEU	Dist-2	Dist-3	Prec.	Rec.	F1
ReDial	21.9	3.72	5.24	28.3	23.0	24.0
DCR	21.9	1.91	3.12	48.1	37.7	40.8
KBRD	22.8	4.92	9.21	42.1	33.3	35.9
KGSF	18.6	4.00	5.34	37.7	32.0	33.2
CR-Walker	26.6	21.2	48.1	46.3	60.3	47.7
-depth=1	28.0	19.2	40.8	50.0	47.7	45.1

Table 3: Automatic evaluation of generation tasks on *ReDial*.

Model	Recommendation				Gen.
	T@1	T@3	C@1	C@3	BLEU
BERT	25.5	66.3	26.4	68.3	23.9
R-GCN+GPT	75.6	95.7	76.1	96.3	25.2
GoRecDial	77.8	97.1	78.2	97.7	27.4
CR-Walker	81.1	97.5	81.3	98.1	29.6

Table 4: Automatic evaluation on *GoRecDial*, where users chat in the context of their watching history.

5.1 Automatic Evaluation

The results on Redial and GoRecDial are shown in Table 2, 3 and 4. As can be seen, CR-Walker outperforms most baselines in both item recommendation and response generation.

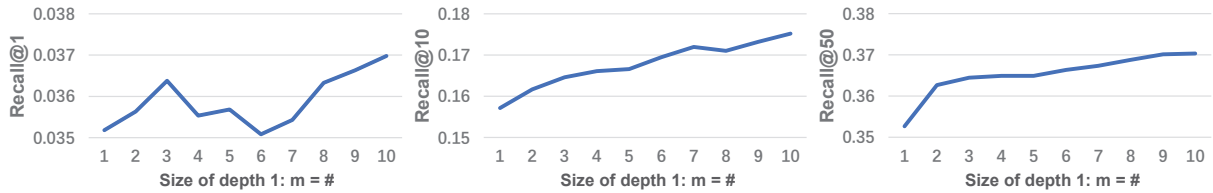


Figure 3: CR-Walker’s recommendation performance with regard to the number of selected nodes at the first hop during reasoning. Most metrics improve as more supporting entities are allowed to be selected.

Item Recommendation We evaluate CR-Walker on item recommendation quality in different settings using metrics proposed in the original datasets: In Redial, we adopt *Recall@k* for evaluation since there are multiple movies recommended in a dialog. In GoRecDial, since the ground-truth movie to recommend is annotated in each dialog, we evaluate the *hit rate* among top-k recommendation at each turn ($T@k$), and also the hit rate *only at the end of* each dialog ($C@k$) to observe the usefulness of conversation further. On Redial, we also use *Coverage* to evaluate recommendation diversity, which is calculated by the proportion of candidate items recommended on test set.

In Table 2, we can find that CR-Walker **outperformed all baselines using a single KG on recommendation quality**, including ReDial, DCR and KBRD. This indicates use of multi-path reasoning can more effectively utilize background knowledge. KGSF uses an additional KG from ConceptNet (Speer et al., 2017) compared with ours, and performs slightly better on *Recall*. However, CR-Walker can obtain a performance gain as well by incorporating ConceptNet as additional feature (+ConceptNet in Table 2), and even outperforms KGSF on *Recall@1* and *Recall@10*, but this is not the focus of this paper. Regarding recommendation diversity, CR-Walker outperformed all baselines including KGSF. The tree structured reasoning enables multiple items to be recommended at the second hop, each with its certain attributes related to earlier conversation. This results in a higher coverage of candidate items compared with 1-hop reasoning that directly arrives at recommendation.

In Table 4, we can find that CR-Walker obtains the **best performance on all recommendation metrics if the user has a clearer preference**. Surprisingly, we also find that $T@1$ is close to $C@1$ in CR-Walker in GoRecDial. This is because entity embedding provides overly strong information to distinguish the correct movie from only five candidates so that it can offer good recommendations

easily, even without user utterances.

Response Generation We apply *BLEU* and *Distinct-n* (Li et al., 2016) to measure the generated response on word-level matches and diversity. Noting that different from Chen et al. (2019) that calculate sentence-level Distinct, we use corpus-level Distinct to give a more comprehensive assessment. Following Wu et al. (2019), we also adopt *knowledge F1-score* to measure knowledge exploitation. Unlike metrics in item recommendation, the knowledge score is calculated by corresponding generic classes rather than the exact match. For example, it only evaluates whether the system mentioned the *genre* to promote movie recommendation but does not care about the exact genre.

Results show that CR-Walker outperforms all baselines on corpus-level **language diversity by a large margin** (dist-2,3 in Table 3). Noticeably, while CR-Walker achieves the highest BLEU in GoRecDial, BLEU in ReDial drops a little when incorporating tree-structured reasoning into the response generation process (26.6 vs. 28.0). This is because CR-Walker sometimes infers different reasoning trees, and afterwards generates sentences that differ from the references but may also be reasonable. We resort to human evaluation (Sec. 5.3) to further evaluate the language quality.

In addition, CR-Walker obtains the best knowledge recall and F1 scores. This indicates that CR-Walker **reasonably utilizes informative entities** during conversational recommendation. A slightly lower precision in CR-Walker is also because it produces different reasoning trees.

5.2 Ablation Study

To understand CR-Walker’s superiority against other baselines, we further examine the influence of tree-structured reasoning on the recommendation performance. We first study the effect of **tree depth**. When we simplify the reasoning process by removing the first hop reasoning and force the model to directly predict the entities at the second

hop (-depth=1 in Table 2), there is an apparent decline in all Recall@k. R-GCN+GPT shares a similar framework with CR-Walker-depth=1 which directly recommends items using R-GCN, and CR-Walker outperforms it much on item recommendation. These results demonstrate that **two-hop graph reasoning** better exploits the connection between entities by exploring intermediate entities, and it is crucial for successful recommendation.

We then study the effect of **tree width**. We control the width of reasoning paths by setting the maximum number of entities m allowed to be selected at the first hop, and observe the performance in Recall@k, as shown in Fig. 3. Overall, the performance of CR-Walker increases as m goes up. Though there is a slight decrease in Recall@1 when the width grows to around 6, the performance gains in the end. This can be interpreted as **multi-path reasoning** is superior to single-path reasoning by providing the model with multiple guidance to arrive at the final recommendation.

5.3 Human Evaluation

In addition to automatic evaluation, we conduct point-wise human evaluation. 300 posts are randomly sampled from the test set. For each response generated by each model, we ask 3 worker from AMT to give their ratings according to each metric with a 3-point scale (3/2/1 for good/fair/bad respectively). The average score of each metric is reported. Among the metrics, *fluency* and *coherence* focus on the response generation quality, *informativeness* and *effectiveness* evaluate whether the conversation is well-grounded in a recommendation scenario. In particular, *informativeness* evaluates whether the system introduces rich movie knowledge, and *effectiveness* evaluates whether the system engages users towards finding a movie of interest successfully.

We present human evaluation results on ReDial in Table 5. We adopt GPT-2 as an additional baseline fine-tuned on the training set and generates response directly. We find that it serves as a solid baseline due to the success of PLMs in language generation and incorporating knowledge implicitly. Although GPT-2 cannot make actual recommendation since it does not “select” a movie, it outperforms all the previous baselines even on *informativeness* and *effectiveness*. This implies that finding the appropriate recommendation is insufficient to satisfy users under the conversational rec-

Model	Fluency	Coherence	Inform.	Effect.
ReDial	2.31	1.96	1.69	1.74
DCR	2.12	1.84	1.68	1.63
KBRD	2.45	2.14	1.95	1.89
KGSF	2.17	1.96	1.98	1.93
GPT-2	2.47	2.24	2.05	1.98
Human	2.52	2.34	2.18	2.10
CR-Walker	2.60*	2.41*	2.33*	2.22*

Table 5: Human evaluation on *ReDial*. *Human* responses come from the ground truth responses provided in the dataset. Numbers marked with * indicate that the improvement is statistically significant (t-test with p-value < 0.05).

Model	Fluency	Coherence	Inform.	Effect.
Human(+)	2.54	2.38	2.26	2.15
CR-Walker(+)	2.61	2.41	2.33	2.23
Human(-)	2.46	2.23	1.99	1.98
CR-Walker(-)	2.57	2.41	2.33	2.18

Table 6: Human vs. CR-Walker. (+) and (-) indicate the subset of responses where two competitors share the same intent and pose different intent respectively.

ommendation setting, but the quality of natural language may also determine how well recommendations will be accepted. CR-Walker, equipping the PLM with external knowledge and reasoning ability, further boosts GPT-2’s performance by providing interpretable recommendation through utterance. Among all the metrics, CR-Walker improves informativeness and effectiveness more significantly. We observe that CR-Walker can generate utterance with more detailed attribute information to support recommendation compared to GPT-2 alone. This demonstrates that CR-Walker succeeds in generating engaging responses with tree-structured dialog acts beyond PLMs.

We further study why CR-Walker can outperform human responses. In terms of the system action, the intent accuracy of CR-Walker reaches only 67.8%, but we find that a different intent from the human’s choice sometimes results in better informativeness and effectiveness. We calculate the score separately for humans and CR-Walker based on whether the intent selection is the same or different in Table 6. For identical intents, CR-Walker’s improvements on four metrics are all marginal, as the improvement only comes from providing more information at the first hop reasoning. For different intents, however, the human performance drops remarkably, while our performance remains consistent. We observe several samples and find that

the human usually performs perfunctory chit-chat like “haha” or “lol” in these cases. By contrast, CR-Walker replies with a relevant query or appropriate recommendation⁶. This implies that the score advantage may come from the explicit reasoning on system actions that CR-Walker learns.

5.4 Recommendations in Dialog Flow

We also analyze the flow of recommended items throughout conversation among various interaction cases, where we roughly categorize the flow into two patterns. In one pattern, the seeker chats around a fixed topic of interest and ask for similar recommendations. This pattern is common on Redial, and CR-Walker efficiently handles it by making appropriate recommendation through tree structure reasoning. However, in a less common case where user suddenly change to a new topic, earlier recommendations would have little effect on the latter ones. In these cases, reasoning on previous items may result in inappropriate recommendations. In practice, we weigh the two patterns by setting the maximum length of dialog history l_{max} , where we only used last l_{max} utterances in D to compute utterance embedding and user portrait. When we set $l_{max} = 3$, we empirically find CR-Walker can handle most topic changes while still providing appropriate recommendation during interaction.

6 Conclusion and Future Work

We have presented CR-Walker, a conversational recommender system that applies tree-structured reasoning and dialog acts. By leveraging intermediate entities on the reasoning tree as additional guidance, CR-Walker better exploits the connection between entities, which leads to more accurate recommendation and informative response generation. Automatic and human evaluations demonstrate CR-Walker’s effectiveness in both conversation and recommendation. It is worth noting that the dialog acts used in CR-Walker are automatically obtained by entity linking to a KG with simple heuristics. Therefore, our work can be easily applied to different conversational recommendation scenarios.

There are still some topics to be explored based on CR-Walker. It would be promising to equip CR-Walker with a language understanding module to capture users’ negative feedback and propose other reasoning rules to handle such situations. An

⁶The case study is provided in the appendix.

efficient way to learn reasoning more than two hops can be investigated in the future as well.

Acknowledgement

This work was supported by the National Science Foundation for Distinguished Young Scholars (with No. 62125604) and the NSFC projects (Key project with No. 61936010 and regular project with No. 61876096). This work was also supported by the Guoqiang Institute of Tsinghua University, with Grant No. 2019GQG1 and 2020GQG0005.

References

- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.
- 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.
- 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*, pages 139–148.
- 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*, pages 815–824.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text de-generation. In *Proceedings of the 8th International Conference on Learning Representations*.
- Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2020. A survey on conversational recommender systems. *arXiv preprint arXiv:2004.00646*.
- Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul A 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.
- 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 *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 304–312.
- 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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2073–2083.
- 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.
- 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*, pages 9725–9735.
- 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.
- Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.
- 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 *Proceedings of the 15th International Conference on Extended Semantic Web Conference, ESWC 2018*, pages 593–607. Springer/Verlag.
- 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*, pages 4444–4451.
- Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 235–244.
- Ryuichi Takanobu, Runze Liang, and Minlie Huang. 2020. Multi-agent task-oriented dialog policy learning with role-aware reward decomposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 625–638.
- Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. 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 100–110.
- Dai Hoang Tran, Quan Z Sheng, Wei Emma Zhang, Salma Abdalla Hamad, Munazza Zaib, Nguyen H Tran, Lina Yao, and Nguyen Lu Dang Khoa. 2020. Deep conversational recommender systems: A new frontier for goal-oriented dialogue systems. *arXiv preprint arXiv:2004.13245*.
- David R Traum. 1999. Speech acts for dialogue agents. In *Foundations of rational agency*, pages 169–201. Springer.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.
- Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804.
- Hu Xu, Seungwhan Moon, Honglei Liu, Bing Liu, Pararth Shah, and S Yu Philip. 2020. User memory reasoning for conversational recommendation. In

Proceedings of the 28th International Conference on Computational Linguistics, pages 5288–5308.

Tong Yu, Yilin Shen, and Hongxia Jin. 2019. A visual dialog augmented interactive recommender system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 157–165.

Ruiyi Zhang, Tong Yu, Yilin Shen, Hongxia Jin, and Changyou Chen. 2019. Text-based interactive recommendation via constraint-augmented reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 15214–15224.

Xiaoying Zhang, Hong Xie, Hang Li, and John CS Lui. 2020. Conversational contextual bandit: Algorithm and application. In *Proceedings of The Web Conference 2020*, pages 662–672.

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*, pages 177–186.

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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1006–1014.

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.

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*, pages 881–890.

A Notation

Notations used in this paper are summarized in Table 7.

Notation	Description
e, E	Entity, entity set
r, R	Relation, relation set
t	Dialog turn
N_e^r	Neighbors of e in relation r
$Z_e^{(n)}$	Neighbors of e at n -th hop reasoning
h_e	Entity embedding of e
u_t	Utterance embedding
p_t	User portrait
$E_t^{(n)}$	Entity set at n -th hop reasoning
A	Dialog act

Table 7: Notations used in the CR-Walker.

B Pseudocode

The entire reasoning and training process of CR-Walker is described in Algorithm 1.

C Implementation Details

In experiments, we train the model on a single Tesla-V100 GPU with a learning rate of 1e-3, batch size of 36, and max epoch of 60. *Adam* is used as the optimization algorithm, with a weight decay of 1e-2. We set the max number of selection at the first hop $m = 5$ during training, and used negative sampling for the candidate items (second hop when system intent is recommend). The ratio between negative and positive samples is set to 5. The dimension of entity embedding d is set to 128. The layer size of R-GCN L is set to 1. BERT-base and GPT-2-medium are applied from Wolf et al. (2020) and the parameters of the BERT encoder are frozen during the training process. The weights of graph walker loss at each hop are $\lambda_1 = 1, \lambda_2 = 0.1$ for GoRecDial and $\lambda_1 = 1, \lambda_2 = 1$ for Redial, respectively. During inference, we apply $\tau = 0.5$ as the entity selection threshold and $p = 0.9$ for the response decoding strategy. Bag of words (BOW) of the movie description are encoded using a fully connected layer as additional features in GoRecDial.

During KG construction, the generic classes we introduce are the *director, actor, time, genre, subject* related to each movie. All entities are directly extracted from DBpedia, except for genres, which are taken from MovieLens. There are 12 types of relationships between entities, namely *actor of / director of / genre of / subject of / time of / is a*

Algorithm 1: Conversational Recommendation Walker

Input: Knowledge graph G , training data D

```
1 Initialize the parameters of intent classifier
   $\theta_i$ , walker cell  $\theta_w$  and generation model  $\theta_g$ 
2 for  $j = 1$  to  $|D|$  do
3   Set all the entities on  $G$  “unmentioned”
4   for  $t = 1$  to  $T_j$  do
5     // Intent selection
6     Obtain utterance embedding  $u_t$  w/
7     Eq. 1
8     Calculate  $L_{int}$  based on Eq. 4
9     // Graph reasoning
10    Obtain user portrait  $p_t$  w/ Eq. 3
11    Update the entities mentioned in
12    user utterances on  $G$  as
13    “mentioned”
14    Set  $E_t$  as singleton of the selected
15    intent
16    for Hop  $n = 1$  to 2 do
17      Select entities using all elements
18      in  $E_t$  w/ Eq. 7
19      Calculate  $L_i$  at current hop w/
20      Eq. 9
21      Update the selected entities on
22       $G$  as “mentioned”
23      Set  $E_t$  as all selected entities at
24      current hop
25    end
26    // Response generation
27    Transform the reasoning tree into
28    the dialog act  $A$ 
29    Calculate  $L_{gen}$  w/ Eq. 10
30    Perform gradient descent on  $L$  w/
31    Eq. 11
32  end
33 end
```

and their corresponding reverse relationships. The last relationship *is a* and its reverse appears between attributes and generic classes, while other relationships appear between attributes and candidate items. We then perform entity linking in each sentence, and use regular expressions to determine its intent and entities on the reasoning tree. Noting that no annotation is available in two datasets, and we label them in an automatic way. Dataset statistics are shown in table 8, while detailed entity and relation statistics for KG are listed in table 9.

Dataset	Dialogs	Utterances	Items	Entities	Relations
GoRecDial	9K	171K	3.8K	19.3K	227K
ReDial	10K	182K	6.9K	30.5K	393K

Table 8: Dataset statistics.

Dataset		GoRecdial	Redial
Entities	Movies	3,782	6,924
	Persons	7,936	12,803
	Subjects	7,558	10,707
	Genres	18	18
	Times	7	12
	Generals	7	7
	total	19,308	30,471
Relations	actor_of	16,472	27,639
	director_of	3,634	6,063
	genre_of	8,386	18,259
	subject_of	61,940	107,356
	time_of	3,782	6,309
	is_a	19,478	30,715
	total	227,384	392,682

Table 9: Knowledge graph statistics of **GoRecdial** and **Redial**. The total relations are twice the sum of 6 types of relationships listed on the table due to the addition of corresponding reverse relationships.

D Case Study

We finally present an interactive case here to demonstrate our model’s capability during interactive dialog and our model’s explainable nature. The sequential dialog acts corresponding to the reasoning tree generated by CR-Walker is presented in Table 10 along with the dialog. We mark all the mentioned entities either in bold (user turn) or in colors (system turn) according to the reasoning hop. The dialog starts with greetings between the user and CR-Walker, followed by CR-Walker proactively seeking user preference by asking which kind of movie he or she likes. The following few turns focus on recommending action movies, and CR-Walker provides an appropriate description of the recommended movies and some comments on *Arnold Schwarzenegger*’s muscles. The topic then switches to horror movies after the user explicitly requires scary ones, with the system recommending four appropriate movies within two turns. The dialog finally ends with the user expressing gratitude and CR-Walker expressing goodwill. Overall, at the utterance level, the whole dialog contains appropriate amounts of information and various dialog acts from the model, enabling the conversation to appear coherent and fluent.

The intermediate dialog acts that CR-walker gen-

U: Hi there, good evening to you!
D: [Query (Genre)]
R: Good evening to you also! What **kind of films** do you like?
U: I'm looking for some **action** flicks like **Lethal Weapon** to watch with my friends tomorrow. Any ideas?
D: [Recommend (Action (The Terminator)) (1980s (The Terminator))]
R: Okay one sec. what about **The Terminator**. It's **old** but really great **action** flick.
U: **Terminator**'s great! **Arnold Schwarzenegger** is a beast in it. We have watched it several times though.
D: [Chat (Action (Die Hard))]
R: Indeed he is, all that muscle. I would also recommend **Die hard** then. It's a really good mixed **action** movie.
U: Great. Can you also give me some **scary** ones like **Halloween** or **Annabelle**?
D: [Recommend (Horrow (Shining) (It))]
R: Oh yes! Have you seen **The Shining** or **It**?
U: Yeah... those are really good. We watched **The Shining** one time and had trouble sleeping...
D: [Recommend (Horrow (Split) (Saw))]
R: **Split** and **Saw** are even more on the **scary** side!
U: Great, I'll check these too. Thanks a lot for the help!
D: [Chat]
R: No problem. Have a great day!

Table 10: A sampled case between CR-Walker (R) and a real user (U). The sequential dialog acts (D) at each dialog turn derived from the reasoning tree are presented as well. Entities mentioned in user's turn are marked in **bold**, while entities in system's turn are marked in colors based on the number of reasoning hop (blue-hop 1, purple-hop 2)

erates help us to better control and understand the generated utterance. On one hand, the entity on the reasoning tree provides additional insight into the model's particular statement. Generated sentences may contain the entity name directly, but may also contain paraphrase of entities, as in cases of *Genre*, *1980s* and *Horror* mapping to *kind of films*, *old* and *scary* respectively. The model also learns to omit some of the entities on the reasoning path based on the dialog context, such as entity *Horror* when the system recommended *Shining* and *It*. Such non-trivial paraphrasing would be hard to interpret in the absence of the reasoning tree. On the other hand, the reasoning tree's structure even gives a hint to the approach our model takes when it mentions an entity. An interesting case happens in the third turn of the dialog when CR-Walker recommends *Die Hard*. The predicted dialog intent appears to be "chit-chat", and *Die Hard* is selected at hop 2 in the reasoning process during inference. As a result, the system talks about the attributes of *Die Hard* (use of *Action*) instead of directly recommending it, and the tone taken by the model is more

casual and relevant to the previous context (use of *then* and comment of *all that muscle*). Together, the above advantages add to our model's explainability, giving our model the edge to be interpreted beyond words.