Bridging the Gap between Conversational Reasoning and Interactive Recommendation

Wenchang Ma[*], Ryuichi Takanobu[*], Minghao Tu, Minlie Huang[†]

Tsinghua University, Beijing, China

Abstract

There have been growing interests in building a conversational recommender system, where the system simultaneously interacts with the user and explores the user's preference throughout conversational interactions. Recommendation and conversation were usually treated as two separate modules with limited information exchange in existing works, which hinders the capability of both systems: (1) dialog merely incorporated recommendation entities without being guided by an explicit recommendation-oriented policy; (2) recommendation utilized dialog only as a form of interaction instead of improving recommendation effectively.

To address the above issues, we propose a novel recommender dialog model: CR-Walker. In order to view the two separate systems within a unified framework, we seek high-level mapping between hierarchical dialog acts and multi-hop knowledge graph reasoning. The model walks on a large-scale knowledge graph to form a reasoning tree at each turn, then mapped to dialog acts to guide response generation. With such a mapping mechanism as a bridge between recommendation and conversation, our framework maximizes the mutual benefit between two systems: dialog as an enhancement to recommendation quality and explainability, recommendation as a goal and enrichment to dialog semantics. Quantitative evaluation shows that our model excels in conversation informativeness and recommendation effectiveness, at the same time explainable on the policy level.

1 Introduction

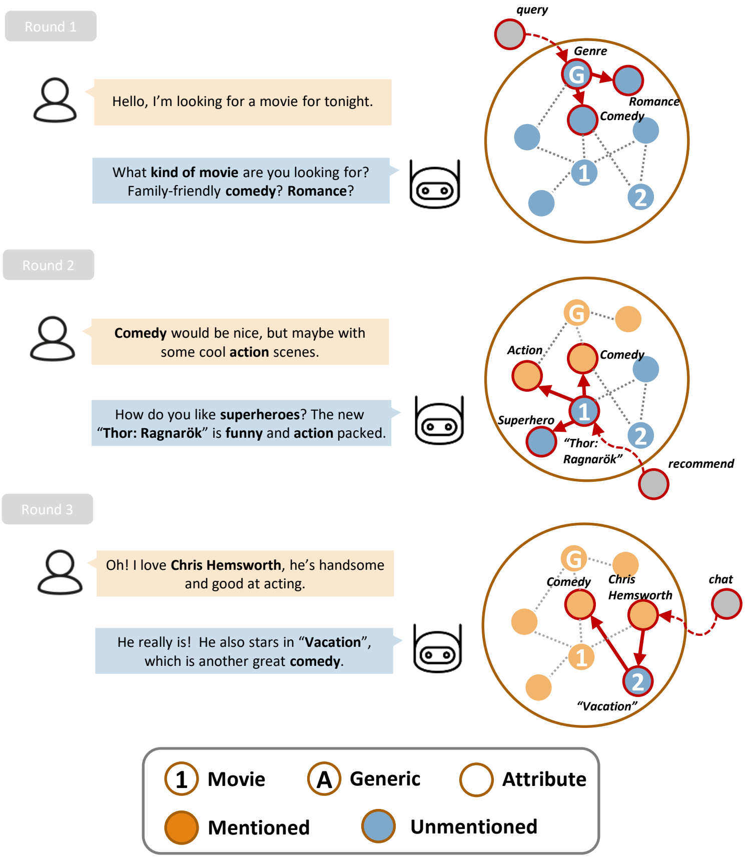
Finding ways to integrate the strength of conversation and existing knowledge with traditional approaches for recommendation has drawn greater research attention [47, 11], but creating a seamless framework that combines dialog, external knowledge, and recommendation still remains challenging. A great part of the challenge lies in that conversational recommender systems should not be a loose coupling of the two separate modules with little information exchange, which usually hinders high-level benefits from each other. Rather, a unified framework should be built after closely examining how the two systems complement.

From the perspective of dialog systems, a recommender dialog can be viewed as a variation of typical task-oriented dialog. Its goal is to provide appropriate recommendations and satisfy the customer. Past studies usually focused on using recommendation knowledge as an enrichment to dialog semantics [3, 49], but neglected recommendation as a goal to guide the high-level dialog policy. While keeping the conversation smooth, the system should plan different strategies into dialog to realize the goal, including seeking user preference proactively, chatting and informing attentively, and providing explanations behind recommendation, etc. In this sense, recommendation should contribute to dialog in both knowledge-aware response generation and recommendation-aware dialog management.

From the perspective of recommender systems, a conversational recommender can be viewed as an enhanced interactive recommender system. In general, these systems were proposed as a solution to the *Cold Start* problem [27, 19] by understanding user interests through user utterances, where traditional recommender systems suffer when no prior knowledge is available. However, there is more to benefit when choosing natural language as the form of interaction. Dialog serves as a powerful tool both in revealing the reasons behind recommendation, rather than a system using a user-item matrix that mainly records the

[*] Both authors contributed equally to this work.

[†] Corresponding author: Minlie Huang.



Hello, I'm looking for a movie for tonight.

What **kind of movie** are you looking for? Family-friendly **comedy**? **Romance**?

Comedy would be nice, but maybe with some cool **action** scenes.

How do you like **superheroes**? The new "**Thor: Ragnarök**" is **funny** and **action** packed.

Oh! I love **Chris Hemsworth**, he's handsome and good at acting.

He really is! He also stars in "**Vacation**", which is another great **comedy**.

Figure 1: First three turns of an example dialog between user and CR-walker. The dialog is shown on the left with entities from the knowledge graph in bold. The graph on each dialog turn's right demonstrates the reasoning process of our model, with the reasoning tree marked red. Throughout this paper, generic classes are noted with letters, recommendation candidates with numbers and other attributes. The orange/blue color indicates that the entity is mentioned/unmentioned in the previous context.

user visit; and in enhancing recommendation effectiveness by using different social strategies, e.g. promoting products persuasively to boost customers awareness. In this sense, aside from seeking user preference, dialog should also contribute to recommendation in both explainability and effectiveness.

To devise a unified framework that can bridge the gap between conversation and recommendation, we propose **C**onversational **R**ecommender **Walker** (CR-Walker), a conversational recommender system that tracks and manages the dynamic conversational recommendation based on a knowledge graph (KG). The key idea is to learn a dialog policy that conducts tree-structured multi-hop reasoning on the KG and also conditioned on the dialog context, which can then be mapped to hierarchical dialog acts to guide generation. A toy example conversation is shown in Figure 1. In each turn of dialog, the agent first determines its general intent based on the conversation history and current graph information, then conducts entity selection hierarchically on each level to form a reasoning tree. Due to the lack of necessary information, the model decides to ask for the user's preference towards different genres in the first turn. After receiving the user's feedback, the model recommends a movie while providing relevant information of why the user might like it. When the user expressed the attitude to a specific actor, the model then casually talked about the actor and another film he starred in. Such a model is conceptually inspiring, acting as both recommendation systems and dialog systems: (1) The model handles the exploration-exploitation of user preferences with a dialog policy, and recommends using various natural language strategies. Actions of each turn are clearly shown on the graph, making the recommendation process explainable. (2) The tree-structured reasoning enables multiple entities with varying hierarchical connections to be selected, which functions as a backbone for generating diverse and informative responses. Results show that CR-Walker outperforms the state-of-the-art models in prior work on two public datasets, as well as in human evaluation.

In brief, our contributions are summarised below: (1) From the dialog aspect, our model readily functions as a recommendation-oriented policy. Compared to previous methods, the model better handles strategical dialog management to conduct conversation towards the goal while explicitly incorporating recommendation-related knowledge to enrich semantics. (2) From the recommendation aspect, our model readily functions as a language-interactive reasoner. Compared to previous methods, it excels in recommendation explainability by tracking the graph reasoning process while enhancing recommendation effectiveness using natural language. (3) We demonstrate our model's capability on two conversational recommendation datasets, achieving state-of-the-art performance on most automatic metrics. Human evaluation further demonstrates that our model can generate informative and effective responses comparable to human performance on turn level. Our proposed model reveals the new potential for knowledge-based conversational recommender systems.

2 Related Works

2.1 Conversational Recommendation

Conversational Recommender Systems (CRS) learn and model user's preferences through interactive dialog conversations, which supports a richer set of user interaction in recommendation [11]. Previous work on CRS can be categorized into two types.

One is dialog-based CRS [17, 12, 18, 21] that make recommendations using free text, which have much flexibility to influence how the dialog continues. As these systems suffer from existing limitations in natural language processing, e.g. understanding user preferences expressed in natural language, most methods incorporate external information such as knowledge base and user logs to enhance the dialog semantics [44, 49] or update the user representations [45, 3]. However, these methods neither fully utilize relations between items and their attributes to generate informative responses nor capture higher-level strategic behaviors in recommendation to guide the conversational interaction.

The other is recommendation-based CRS [31, 47, 46, 50] that ask questions about user preference over pre-defined slots or attributes to select items for recommendation. This has been attractive to developers as the actions available to users are clear and controllable. As the system's answers can be grouped into some pre-defined intents, and then can be implemented with the help of template snippets, most approaches of this type do not address response generation [14, 15]. Under this simplified setting, such dialogs can only provide limited actions (e.g. ask on a fixed set of attributes [5, 4]) without explicit explanation why the system makes such recommendation, thus the users might feel restricted or unnatural in the way the system makes interactive recommendations.

2.2 Task-Oriented Dialog Policy

CRS can be regarded as a variation of Task-oriented Dialog Systems (TDS) that supports its users in achieving recommendation-related goals through a multi-turn dialog [35]. User goals are explicit and clear in traditional TDS, whereas user interests are implicit and ambiguous in CRS. As both TDS and CRS aim to meet user demands via multi-turn interaction, we can draw some techniques from TDS to do better planning and lead the dialog during recommendation.

As a part of dialog management, dialog policy [30, 32] plays a vital role in TDS because it decides the next action that the agent should take during conversation. Numerous studies [48, 2, 25, 33, 7] have been proposed in recent years to train an effective dialog policy. The action space for those methods is usually the set of *dialog acts* [36, 29, 24], an abstract representation of speaker's intention.[1] Using dialog acts can reduce dialog policy's output space from a large word vocabulary (around 10k) to a small number of semantic intents (less than 100). Therefore, it is promising to choose a set of domain-independent dialog acts to facilitate information filtering and decision making in conversational recommendation.

[1] For example, the sentence "Find me a Japanese restaurant near the Central Park" can be transformed into a dialog act: {domain=restaurant, intent=inform, [slot=food, value=Japanese; slot=location, value=Central Park]}.

Table 1: Notations used in the CR-Walker.

Notation	Description
$e, \mathcal{E}$	Entity, entity set
$r, \mathcal{R}$	Relation, relation set
t	Dialog turn
$\mathcal{N}_e^r$	Neighbors of e in relation r
$\mathcal{N}_e^{(i)}$	The i-th hop neighbors of e
$\boldsymbol{h}_e$	Graph embedding of e
$\boldsymbol{u}_t$	Utterance embedding
$\boldsymbol{p}_t$	User portrait
$\mathcal{E}_t^{(i)}$	Entity set at tree depth i
$\mathcal{T}$	Reasoning tree
A	Dialog act

2.3 Knowledge Graph Reasoning

Recently, introducing a knowledge graph (KG) into the recommender system as side information has attracted much research attention. A KG is a heterogeneous graph, where nodes function as various entities including items and attributes, and edges represent relations between entities. Some studies utilize the KG information by learning entity embeddings to enrich the representations of items and attributes [40, 10], or leveraging the connectivity patterns of the entity in the graph for additional guidance [9, 43], and other works introduce graph neural networks [13, 39] for multi-hop graph reasoning in recommender systems [41, 34]. Compared with traditional recommender systems, KG-based recommendation makes the reasoning process available.

Recently, knowledge graph reasoning has been applied in dialog systems as well [37, 23, 22]. Graph reasoning is essential for generating dialog policy from a KG, where the graph can be viewed as a structured form of state representation. However, these works mainly focus on the knowledge selection and response generation, not on the user-item interaction, which is essential in a recommendation scenario.

3 Conversational Recommender Walker

In this section, we introduce our proposed CR-Walker framework. Figure 3 shows the overall architecture of our model: dialog context $\boldsymbol{u}_t$ and user portrait $\boldsymbol{p}_t$ obtained from graph embedding of mentioned entities are used to guide the tree-structured walk on the graph. A mentioned entity set is dynamically updated during the interaction to maintain the status of each entity. The reasoning tree produced is then transformed into dialog acts to guide the response generation. We start by addressing our model's key intuition: how to map dialog acts to multi-step reasoning on the KG. Guided by this concept, we further discuss the formulation of different modules in detail. Notations used in this paper are shown in Table 1.

3.1 From Reasoning Tree to Dialog Act

We define a typical knowledge graph $G = (\mathcal{E}, \mathcal{R})$ under recommendation settings as follows: the entities on the graph are divided into three categories, namely the *recommendation candidates*, *attribute features*, and *generic classes*. Each candidate entity is related to a set of attribute features, while each attribute feature is connected to its corresponding generic class. There might also exist relationships between different attributes. To provide a more concrete example, under the movie recommendation setting, the candidate movie *Titanic* is linked to attribute features *Romance*, *Leonardo DiCaprio* and *James Cameron*, and these three attributes are linked to generic classes *Genre*, *Actor* and *Director*, respectively.

A graph described above now contains all the structured properties useful for making recommendations. At each dialog turn, the recommender's action can be viewed as navigating on such a graph to select informative entities to keep the conversation engaging and meaningful. However, the large search space for entities on a large-scale KG makes direct selection infeasible. Previous work [23] had tried to tackle this problem by looking for a single reasoning path that originates from the entities mentioned in the last turn.

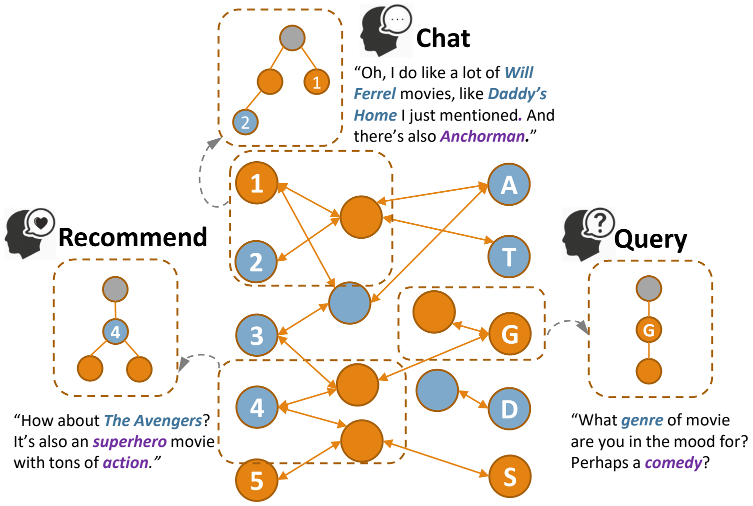

Chat
"Oh, I do like a lot of Will Ferrel movies, like Daddy's Home I just mentioned. And there's also Anchorman."
Recommend
"How about The Avengers? It's also an superhero movie with tons of action."
Query
"What genre of movie are you in the mood for? Perhaps a comedy?"
1
2
A
T
3
G
4
D
5
S

Figure 2: Illustration of graph formulation and mapping between reasoning tree and dialog acts. In each dialog turn, CR-Walker expands a reasoning tree from the current dialog act of recommend/query/chat (depth 0) to recommendation candidates/generic classes/mentioned entities (depth 1), respectively. The expansion continues to include more relevant neighboring entities (depth ¿ 1). Entities on the reasoning tree are later transformed to informative words within generated utterance.

Whereas the reasoning process within conversational recommendation is much more complicated: the origin of reasoning may not be from the last turn, and the reasoning path is usually not single-threaded. Thus, a feasible yet more expressive approach to entity selection is crucial for dialog generation.

Inspired by the use of dialog acts to abstract semantics and discourse in TDS, we further observe the dialog act's connection with the graph reasoning process. We found that the same intent usually centers on key entities with similar features, with relevant neighboring entities added to elaborate. This resembles the hierarchical selection process of slot and values, forming a tree structure on the graph instead of a single path. We therefore propose to better model the reasoning process **using a reasoning tree that spans from a set of intent-related start points**.

Figure 2 shows the formulation of our KG and the mappings between the reasoning tree and generated utterance guided by dialog acts. We summarize three different intents under the recommender dialog setting to represent common strategies used in CRS: query, recommendation, and chat. The mapping process is formalized below, where we assume the root of reasoning tree the intent itself and entities at depth 1 and deeper depths marked in blue and purple respectively: (1) Dialog intent of **query** maps to the set of generic classes. The reasoning tree first expands to several generic classes like genres and actors to query, then further expands to more specific attribute entities. This corresponds to the proactive behavior of a recommender, where it seeks the user's preference at different levels of abstraction. (2) Dialog intent of **recommendation** maps to the set of recommendation candidates. The reasoning tree first expands to candidates to recommend, then further expands to attributes of each candidates. This corresponds to the recommender making recommendations then back-tracing the reasons why the user might like them. (3) Dialog intent of **chit-chat** maps to the set of mentioned entities. The reasoning tree first expands to several mentioned entities, then further expands to all other neighboring entities. This corresponds to the various skills of a recommender to carry on the dialog naturally and smoothly.

3.2 Knowledge Graph Representation

To introduce external structured knowledge, we extract graph from DBpedia [1] and add generic classes in the fashion introduced in Sec. 3.1. We encode the graph using R-GCN [28], for its capability of modeling neighboring connection more accurately by considering different relations. Formally, for each entity $e \in \mathcal{E}$,

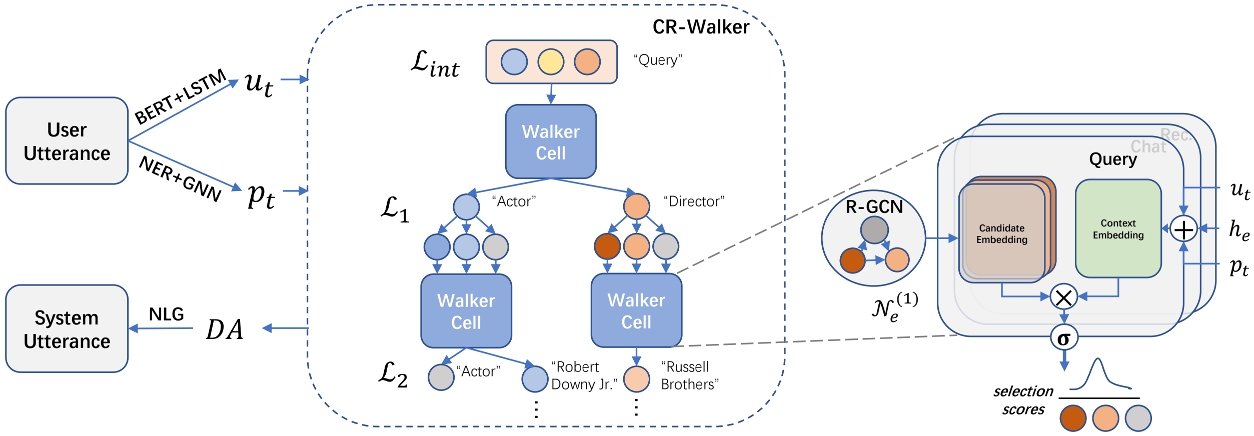


Figure 3: Left: Illustration of the overall architecture for our model. The graph walker performs tree-structured reasoning on the knowledge graph, which is then transformed into dialog acts. Right: Detailed structure for a single walker cell. A walker cell calculates the similarity between the entities on a graph and the context embedding that integrates current utterance embedding and user portrait. The selection process of each entity is learned by logistic regression to enable multiple selections.

the graph embedding $\boldsymbol{h}_e^{(l)} \in \mathbb{R}^{d^{(l)}}$ at each layer l is calculated as:

$$\boldsymbol{h}_e^{(l+1)} = \sigma(\sum_{r \in \mathcal{R}} \sum_{e' \in \mathcal{N}_e^r} \frac{1}{c_{e,r}} \boldsymbol{W}_r^{(l)} \boldsymbol{h}_{e'}^{(l)} + \boldsymbol{W}_0^{(l)} \boldsymbol{h}_e^{(l)}), \tag{1}$$

where $\mathcal{N}_e^r$ denotes the set of neighboring entities of e under the relation r, $c_{e,r}$ is a normalization factor, and $\boldsymbol{W}_r^{(l)}$,$\boldsymbol{W}_0^{(l)} \in \mathbb{R}^{d^{(l+1)} \times d^{(l)}}$ are learnable matrices for integrating relation specific message from neighbors and the current layer's features respectively. The initial features $\boldsymbol{h}_e^{(0)}$ are learnable parameters initialized from a uniform distribution. At the final layer L, the R-GCN embedding $\boldsymbol{h}_e^{(L)}$ is taken as the entity representation, and is denoted as $\boldsymbol{h}_e$ in the following section.

3.3 Dialog History Representation

In a recommender dialog, especially under the cold-start setting where no prior user preference is available, the dialog history is the only way for the recommender to gather information about the user and determine the next step's dialog act. Therefore, obtaining a meaningful representation of dialog history as input to the dialog policy is crucial for performance. We encode the dialog history in two separate ways to reinforce the quality of representation, obtaining the utterance embedding $\boldsymbol{u}_t$ and user portrait $\boldsymbol{p}_t$.

The utterance embedding focuses on the contextual aspect of the dialog history. Specifically, it is obtained through applying a LSTM over the representations of each dialog turn x_t. A dialog turn here refer to a single round of dialog between system and user, where we assume the dialog is initiated by the system. We use a pretrained-BERT [6] to encode each turn and use the embedding of "[CLS]" token as the turn's representation, denoted as $\text{BERT}(x_t)$. The LSTM updates its hidden state as following:

$$\boldsymbol{u}_t = \text{LSTM}(\boldsymbol{u}_{t-1}, \text{BERT}(x_t)), \tag{2}$$

where the hidden state of LSTM $\boldsymbol{u}_t \in \mathbb{R}^{d_u}$, the overall representation of the context, is taken as the utterance embedding. We set the hidden dimension of the LSTM d_u the same as the graph embedding size $d^{(L)}$.

The user portrait, on the other hand, focuses on the informative entities mentioned in the dialog history to learn a representation of user's interests. In each turn, we performed *named entity recognition* to identify informative entities mentioned in the previous user utterance using SPaCy[2], then linked them to entities

[2] https://spacy.io/

on the KG with fuzzy string matching[3]. Identified entities are added to the mentioned entity set. We thus obtain the mentioned embeddings $\boldsymbol{M}_t$ with size of $d^{(L)} \times |M_t|$:

$$\boldsymbol{M}_t = (\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_{|M_t|}).$$

Following [3], we calculated the user portrait via self-attention:

$$\begin{aligned} \boldsymbol{\beta}_t &= \boldsymbol{W}_a^2 \tanh(\boldsymbol{W}_a^1 \boldsymbol{M}_t), \\ \boldsymbol{\alpha}_t &= \text{softmax}(\boldsymbol{\beta}_t), \\ \boldsymbol{p}_t &= \boldsymbol{\alpha}_t * \boldsymbol{M}_t. \end{aligned} \tag{3}$$

Here, both $\boldsymbol{W}_a^1 \in \mathbb{R}^{d_a \times d^{(L)}}$ and $\boldsymbol{W}_a^2 \in \mathbb{R}^{d_a}$ are learnable parameters.

3.4 Heterogeneous Hierarchical Reasoning

As explained in Sec. 3.1, the graph reasoning process to select informative entities is highly complicated. Reasoning can initiate from multiple start points on the graph and extend into multiple paths from each of the start points. To reduce complexity, we assumed that start points share similarities under a certain dialog intent, thus the reasoning process would form a tree with the abstract intent as the root. Our policy aims to simulate the reasoning process described above: select the root (intent), then extend it at different depths to form the final reasoning tree.

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

$$p_{\theta_i}(x_t) = \text{softmax}(\boldsymbol{W}_{int}^2 \text{ReLU}(\boldsymbol{W}_{int}^1 \boldsymbol{u}_t + \boldsymbol{b}_{int}^1) + \boldsymbol{b}_{int}^2). \tag{4}$$

Here $(\boldsymbol{W}_{int}^1, \boldsymbol{b}_{int}^1)$ and $(\boldsymbol{W}_{int}^2, \boldsymbol{b}_{int}^2)$ denote trainable parameters for two fully-connected layers, through which the utterance embedding is mapped down to prediction probability for three different intents. We use standard cross entropy loss, denoted as $\mathcal{L}_{int}$. Note that we only use dialog history representation $\boldsymbol{u}_t$ here, since BERT embedding is already powerful enough for the classification problem. We empirically find that introducing $\boldsymbol{p}_t$ does not promote the final performance for intent classification.

With the root of the tree fixed, the next step is to expand it into a reasoning tree. For this purpose, we design the walker cell, a neural cell shown in Figure 3, that functions at each node on the reasoning tree. Specifically, it performs single step reasoning to select $[0, |\mathcal{N}_e^{(1)}|]$ neighbors to expand the tree from a given entity e. It first integrates the dialog history representation via gate mechanism to obtain context embedding $\boldsymbol{c}_t$:

$$\begin{aligned} \gamma &= \sigma(\boldsymbol{W}_{C/R/Q}[\boldsymbol{h}_e; \boldsymbol{u}_t; \boldsymbol{p}_t]), \\ \boldsymbol{c}_t &= \gamma \cdot \boldsymbol{u}_t + (1 - \gamma) \cdot \boldsymbol{p}_t. \end{aligned} \tag{5}$$

The notion $\boldsymbol{W}_{C/R/Q}$ indicates that the cell owns separate parameters to different intent of chat, recommendation and query, and chooses which one to use according to the selected intent. Note that the gate is aware of the reasoning stand point e as $\boldsymbol{h}_e$ contributes to deriving γ. The context embedding c_t is then used to score each entity e' within $\mathcal{N}_e^{(1)}$

$$\hat{s}_{e'} = \sigma(\boldsymbol{h}_{e'} \cdot \boldsymbol{c}_t). \tag{6}$$

We apply sigmoid function σ here and treat entity selection as a multi-label logistic regression task, which allows multiple entities to be selected. The estimated selection score $\hat{s}_{e'}$ indicates whether the entity is selected for tree expansion. We describe this process of applying a single walker cell for neighbor selection from entity e as a function below:

$$\text{WALK}(e) = \{e' | \hat{s}_{e'} > 0.5, e' \in \mathcal{N}_e^{(1)}\}. \tag{7}$$

With the walker cell described above, it is now possible to expand the tree from its root. Assume the generated tree $\mathcal{T}_t$ has a max depth of m_t at turn t, and owns an entity set of $\mathcal{E}_t^{(i)}$ at tree depth i. Starting

[3] https://pypi.org/project/fuzzywuzzy/

from $\mathcal{E}_t^{(0)}$, which contains only an abstract root node that represents the selected intent, we dynamically initiate $|\mathcal{E}_t^{(i)}|$ walker cells at the current depth with shared parameters to obtain entities at the next depth:

$$\mathcal{E}_t^{(i+1)} = \bigcup_{e \in \mathcal{E}_t^{(i)}} \text{WALK}(e). \tag{8}$$

The tree expansion stops when no neighboring entity is selected, namely:

$$\mathcal{E}_t^{(m_t+1)} = \emptyset.$$

All entities selected on the generated reasoning tree are then added to the mentioned entity set.

3.5 Conditional Language Generation

Having selected the entities on the reasoning tree $\mathcal{T}$, we obtain a tree-structured dialog act A, and then should give system response y conditioned on the available information. We formulate this as a language model. Given the data $\{A, y\}$, the goal is to build a statistical model parameterized by θ_g to characterize

$$p_{\theta_g}(y|A) = \prod_{k=1}^{K} p_{\theta_g}(y_k|y_{<k}, A), \tag{9}$$

where $y_{<k}$ indicates all tokens before the k-th token.

To facilitate response generation using language models, we have to convert the dialog act into a sequential structure. As a dialog act of CR-Walker contains an intent and slot-value pairs, and it is arranged in a tree structure, we can serialize the dialog act into a flat sequence of tokens in the same way that a parser serializes a tree into a string. For example, a dialog act A has an intent I, which can be regarded as the root. The root has two child nodes (n_0, n_1), where n_0 has two leaf nodes (n_{00}, n_{01}), and n_1 has one leaf node (n_{10}). Then the dialog act A can be parsed into the following format:

$$A = [I(n_0(n_{00})(n_{01}))(n_1(n_{10}))]. \tag{10}$$

Therefore, the hierarchy of A can be represented by the brackets in this way. Obviously, every tree-structured dialog act can be serialized into this format, and every sequence that follows this format can be restored to the original dialog act again.

In this paper, we employ the Transformer framework [38] to parameterize the conditions in Eq. 9, where the model first encodes the sequential dialog act A, and next decodes the response y in an auto-regressive generation process. Inspired by the recent success of the huge pre-trained models, we inherit GPT-2 [26], which is a language model pre-trained on extremely massive text data OpenWebText, in our training process of response generation. During inference, top-p sampling [8] are used for response decoding.

3.6 Model Optimization

We train parameters of walker cells θ_w at each depth i using standard logistic regression loss:

$$\mathcal{L}_i = \sum_{e \in \mathcal{E}_t^{(i-1)}} \sum_{e' \in \mathcal{N}_e^{(1)}} -s_{e'} \log(\hat{s}_{e'}) - (1 - s_{e'}) \log(1 - \hat{s}_{e'}), \tag{11}$$

where $s_{e'} \in \{0, 1\}$ is the label for selecting a neighboring entity or not, and $\mathcal{E}_t^{(i-1)}$ denotes the extracted entity set at dialog turn t at depth $i - 1$. Training the generation model is performed via maximizing the log-likelihood (MLE) of the conditional probabilities in Eq. 9 over the user utterance:

$$\mathcal{L}_{gen} = \sum_{k=1}^{K} \log p_{\theta_g}(y_k|y_{<k}, A). \tag{12}$$

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

Algorithm 1: Conversational Recommender Walker

```
Input: Knowledge graph $G$, training data $D$
Initialize the parameters of intent classifier $\theta_i$, walker cell $\theta_w$ and generation model $\theta_g$
for $n = 1$ to $|D|$ do
    Set all the entities on $G$ "unmentioned"
    for $t = 1$ to $T_n$ do
        // Intent selection
        Obtain utterance embedding $u_t$ w/ Eq. 2
        Calculate $\mathcal{L}_{int}$ based on Eq. 4
        // Graph reasoning
        Obtain user portrait $p_t$ w/ Eq. 3
        Update the entities mentioned in user utterances on $G$ as "mentioned"
        Set $\mathcal{E}_t$ as singleton of the selected intent
        foreach tree depth do
            Select entities adjacent to entities in $\mathcal{E}_t$ w/ Eq. 7
            Calculate $\mathcal{L}_i$ at current depth w/ Eq. 11
            Update the selected entities on $G$ as "mentioned"
            Set $\mathcal{E}_t$ w/ Eq. 8
        end
        // Response generation
        Map the reasoning tree to the dialog act $A$
        Calculate $\mathcal{L}_{gen}$ w/ Eq. 12
        Perform gradient descent on $\mathcal{L}$ w/ Eq. 13
    end
end
```

We jointly optimize all trainable parameters mentioned above. The final loss for optimization $\mathcal{L}$ is a weighted sum of all losses:

$$\mathcal{L} = \mathcal{L}_{int} + \sum_i \lambda_i \mathcal{L}_i + \mathcal{L}_{gen}. \tag{13}$$

The entire reasoning and training process is described in Algorithm 1.

4 Experimental Setting

4.1 Data

We use two public CRS datasets to verify the effectiveness of CR-Walker. (1) ReDial [17] is collected by crowd-sourcing users on Amazon Mechanical Turk (AMT). Two paired workers are each given a different role, recommender or seeker, in the conversation. At least 4 different movies are mentioned in every conversation. Every movie **mentioned** in the dialog is annotated explicitly. (2) GoRecDial [12] is collected in a similar way using ParlAI. In each dialog, each worker is given a set of 5 movies with a description. The seeker's set represents his 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. The movies for recommendation are all gathered from DBPedia in both datasets. Some statistics about two datasets are presented in Table 2.

4.2 Implementation Details

We construct the knowledge graph separately for GoRecDial and Redial. The generic classes we introduce are the director, actor, time, genre, subject related to each movie. The actor, director, time, and subject are

Table 2: Dataset statistics.

Dataset	Dialogs	Utterances	Items	User Profile
GoRecDial	9k	171k	5.3k	Yes
ReDial	10k	182k	51.7k	No

directly extracted from DBpedia, whereas genres are taken from classification results of MovieLens[4]. The extracted KG for GoRecDial and Redial contains 19,308 and 30,471 entities, respectively. We then perform entity linking to each sentence in the dataset and use regular expression to determine its intent. Since no annotation for the human reasoning chain is available, we are only able to construct a simple reasoning tree on the turn level according to automatic mappings in Sec. 3.1, because a longer sequence introduces lots of noise. We therefore restrict the tree expansion during training and inference to a max depth of 2.

In experiments, we train the model with a learning rate of 1e-3, batch size of 36, and max epoch of 30. *Adam* is used as the optimization algorithm, with a weight decay of 1e-2. The dimension of graph/ utterance embedding is set to 128. The layer size of R-GCN L is set to 1. BERT-base is applied and the parameters of the BERT encoder are frozen during the training process. The weights of graph walker loss at each depth are $\lambda_1 = 1, \lambda_2 = 0.1$ for GoRecDial and $\lambda_1 = 1, \lambda_2 = 1$ for Redial respectively. During inference, we apply $p = 0.9$ 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.

4.3 Metrics

In terms of recommendation task, we adopt *Recall@k* for evaluation in ReDial. While in GoRecDial, we evaluate the Hit rate among top-k recommendation at each turn (*Turn@k*), and we also compute Hit@k only at the end of each dialog (*Chat@k*) to further observe the usefulness of conversation. As for dialog task, *BLEU* is used to compare the alignment between the generated sentences and the ground truth utterances. We apply *Distinct-n* [16] that counts the number of distinct n-gram to measure the response diversity in ReDial. Note that different from previous works that calculate sentence-level Distinct, we use corpus-level Distinct to give a more comprehensive assessment. Following [42], we also adopt *knowledge F1-score* to measure knowledge exploitation during the interactive recommendation. Distinct from metrics in recommendation, knowledge score is calculated at the high-level attributes 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 what genre it is.

As automatic evaluation can sometimes be unreliable, especially in dialog evaluation, we provide point-wise human evaluation results. 300 posts are randomly sampled from the test set. For each response generated by each model, 3 workers are hired from AMT to give their ratings according to each metric with a 3-point scale (3 (good), 2 (fair), 1 (bad)). The average score of each metric by 3 workers and 300 posts is reported. Among all metrics measured, *fluency* and *coherence* focus on the response generation quality, *informativeness* and *effectiveness* evaluate whether recommendation and conversation are closely related. In particular, *informativeness* evaluates whether the response has incorporated rich movie knowledge and good language diversity, and *effectiveness* evaluates whether the recommender has utilized various conversational strategies to promote recommendation.

4.4 Baselines

We have compared the proposed model with several strong approaches in Redial: (1) ReDial [17]: The benchmark model of ReDial corpus that applies an autoencoder recommender, RNN-based NLG and a sentiment prediction model. (2) DCR [18]: *Deep Conversational Recommender* uses a pointer network to incorporate global topic control and GCN-based venue recommendation in response generation. (3) KBRD [3]: *Knowledge-Based Recommender Dialog* enriches user representation with a KG to give responses for recommendation that are consistent with the user's interest.

[4] `https://grouplens.org/datasets/movielens/`

Table 3: Automatic evaluation on *ReDial.*

Model	Recommendation			Generation			Knowledge		
	R@1	R@10	R@50	BLEU	Dist-2	Dist-3	Prec.	Rec.	F1
ReDial	2.3	12.9	28.7	21.9	3.72	5.24	28.3	23.0	24.0
DCR	2.7	14.8	30.6	21.9	1.91	3.12	48.1	37.7	40.8
KBRD	3.0	**16.3**	33.8	22.8	4.92	9.21	42.1	33.3	35.9
CRWalker	**3.1**	15.5	**36.5**	22.2	**21.2**	**48.1**	46.3	**60.3**	**47.7**
-reasoning	2.9	14.9	34.3	**28.0**	19.2	40.8	**50.0**	47.7	45.1

Table 4: Automatic evaluation on *GoRecDial.*

Model	Recommendation				Gen.
	Turn@1	Turn@3	Chat@1	Chat@3	BLEU
BERT	25.5	66.3	26.4	68.3	23.9
RGCN+GPT	75.6	95.7	76.1	96.3	25.2
GoRecDial	77.8	97.1	78.2	97.7	27.4
CRWalker	**81.1**	**97.2**	**80.9**	**97.9**	**29.6**

We also apply several conversation recommendation methods as the baselines in GoRecDial: (1) BERT [6]: A BERT fine-tuned on GoRecDial that 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 [28] for recommendation with a Transformer-based language model [38] for dialog. The movies are scored using similar structures within our walker cell by calculating the dot-product between encoder hidden states and R-GCN embeddings. (3) GoRecDial [12]: The benchmark model of GoRecDial corpus that is trained via multi-task supervised learning and bot-play learning by posing the recommendation as a task-oriented game.

5 Results

5.1 Automatic Evaluation

We first test on Redial, where movie candidates exceed thousands, and many baselines are available. As shown in Table 3, on the task of recommendation, CR-Walker obtains the best performance on Recall@1 and Recall@50, while being slightly weaker than KBRD on Recall@10. Curiously, when we reduce the reasoning process by pruning the reasoning tree to a max height of 1 (*-reasoning* in Table 3), there is a slight decline in all Recall@k, which demonstrates that hierarchical graph reasoning has improved the interactive recommendation. On the task of generation, CR-Walker outperforms all baselines on corpus-level language diversity by a large margin (dist-2,3) since it combines both external knowledge and pretrained language model's generation capabilities. We find that other models tend to produce monotonic responses on the corpus level, which largely limits performance. Additionally, we also find that BLEU drops greatly by incorporating knowledge graph reasoning into the process of generating responses (22.2 vs. 28.0). This is because CR-Walker sometimes proposes different reasoning paths, thus generating sentences with poor alignment with the dataset, but we cannot conclude that CR-Walker generates inappropriate responses [20]. We will resort to human evaluation (Sec. 5.2) to further elaborate the language quality. Finally, CR-Walker obtains the best knowledge recall and F1 scores on the task of knowledge while being slightly lower on precision. Though the knowledge precision shows a rise in some baselines, a clear drop can be seen in knowledge recall, causing a decrease in overall knowledge exploitation. This indicates that CR-Walker has reasonably utilized informative entities during conversational recommendation.

To further examine our model's performance on recommendation, the results on another dataset GoRecDial are then presented in Table 4, where the ground truth movie to recommend is annotated in each dialog.

Table 5: Human evaluation on *ReDial*. CR-Walker outperforms human on turn level evaluation. 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.
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
GPT2	2.47	2.24	2.05	1.98
Human	2.52	2.34	2.18	2.10
CRWalker	**2.60***	**2.41***	**2.33***	**2.22***

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

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

After using external information, RGCN+GPT largely outperforms BERT on the recommendation task. GoRecDial performs better than RGCN+GPT by using the movie description's BOW embedding. Finally, CR-Walker obtains the best performance by joint training with the graph reasoning process. Surprisingly, we also find that turn@1 is higher than chat@1 in CR-Walker. This is because graph embedding and BOW together provide overly strong information to distinguish the correct movie from only five candidates so that it can offer good recommendations easily, even without dialog.

In summary, CR-Walker achieves the best recommendation performance thanks to the unified framework that incorporates conversational reasoning on the KG and tree-structured dialog acts. The results in the two datasets also demonstrate the generalization ability of CR-Walker.

5.2 Human Evaluation

We compare five model's performance and the performance for humans on ReDial in Table 5. We adopt GPT-2 as an additional baseline that is directly fine-tuned on the training set for generation. We find that it serves as a very strong baseline that can outperform all the previous models, even on recommendation-related metrics, as it can look for implicit knowledge to find appropriate recommendations after fine-tuning. This indicates that finding the right recommendation is insufficient to satisfy the seeker under dialog settings, but the quality of natural language largely determines how well recommendations will be accepted. Our model, equipping the pretrained language model with policy, knowledge, and reasoning ability, further boosts GPT-2's performance to surpass human responses from the dataset. Among all the metrics, informativeness and effectiveness improve more significantly. This demonstrates that CR-Walker succeeds in maximizing the mutual benefit between recommendation and conversation.

We further study how our model can improve performance on the policy level. Our model's intent accuracy reaches 67.8%, but a different intent from the human's does not indicate a poor choice. On the contrary, we find that our model can adjust the poor behaviors of humans to obtain better informativeness and effectiveness. We calculate the score separately for human and CR-Walker when 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. For different intents, however, the human performance drops significantly, while CR-walker's performance remains consistent. We observe some sampled cases and find that the human usually performs perfunctory chit-chat like "haha" or "lol" in these cases. At the same time, CR-walker replies with a relevant query or appropriate recommendation.

Table 7: A sampled case between CR-Walker (R) and a real user (U). The sequential dialog acts (D) at each dialog turn derived from graph 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 reasoning tree depth (blue-depth 1, purple-depth 2)

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 to watch with my friends tomorrow. Any ideas?
D:	[Recommend (The Terminator (Action) (1980s))]
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 (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 (Split (Horror)) (Saw (Horror))]
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!

This indicates that the score advantage comes from the explicit dialog policy that our model learns.

5.3 Case Study

We finally present an interactive case here to demonstrate our model's capability during interactive dialog and our method's explainable nature. The sequential dialog acts corresponding to the reasoning tree generated by CR-Walker is presented in Table 7 along with the dialog. We mark all the mentioned entities either in bold (user turn) or in colors (system turn) according to the reasoning tree depth. 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 they like. The following few turns focus on recommendation of action movies, and CR-Walker provides an appropriate description of the recommended movies and some interesting 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 explainability of our model can be further interpreted from the dialog act level. 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. Such non-trivial paraphrasing would be hard to interpret in the absence of the reasoning tree. On the other hand, the structure of the reasoning tree even gives a hint to the approach taken when mentioning 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 depth 2 in the reasoning process during inference. This reasoning tree is distinct from the tree with a "recommend" intent as root and *Die Hard* at depth 1. 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.

6 Conclusion and Future Work

We have presented CR-Walker, a conversational recommender system that learns a recommendation-oriented dialog policy on a knowledge graph. By mapping a tree of reasoning and dialog acts on an abstract level, the proposed framework can enhance the mutual benefit between recommendation and conversation, resulting in improved recommendation and generation performance. Automatic and human evaluations demonstrate our model's effectiveness in both conversation and recommendation. The model also yields considerable explainability as shown in case studies, which is superior to previous end-to-end frameworks. It is worth noting that the dialog acts used in CR-Walker are automatically obtained by performing entity linking to an existing KG with simple heuristics. Thereby our work can be easily applied to different conversational recommendation scenarios.

Nevertheless, at the same time, CR-Walker is restricted to simple reasoning trees due to the lack of annotation. Therefore, one possible attempt is to build a CRS dataset provided with more fine-grained annotation of the reasoning process at both turn and dialog levels for future work. We will then consider policy planning between turns to model the recommender dialog strategy on the dialog level.[5]

References

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[2] Lu Chen, Xiang Zhou, Cheng Chang, Runzhe Yang, and Kai Yu. Agent-aware dropout dqn for safe and efficient on-line dialogue policy learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2454–2464, 2017.

[3] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 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, 2019.

[4] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H Chi. 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, 2018.

[5] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 815–824, 2016.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019.

[7] Gabriel Gordon-Hall, Philip John Gorinski, and Shay B Cohen. Learning dialog policies from weak demonstrations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1394–1405, 2020.

[8] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[5] The code will be released at `https://github.com/truthless11/CR-Walker`

[9] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1531–1540, 2018.

[10] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. Improving sequential recommendation with knowledge-enhanced memory networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 505–514, 2018.

[11] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. A survey on conversational recommender systems. *arXiv preprint arXiv:2004.00646*, 2020.

[12] Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul A Crook, Y-Lan Boureau, and Jason Weston. 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, 2019.

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[14] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 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, 2020.

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

[16] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 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, 2016.

[17] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards deep conversational recommendations. In *Advances in neural information processing systems*, pages 9725–9735, 2018.

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

[19] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. Facing the cold start problem in recommender systems. *Expert Systems with Applications*, 41(4):2065–2073, 2014.

[20] Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, 2016.

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

[22] Zhibin Liu, Zheng-Yu Niu, Hua Wu, and Haifeng Wang. Knowledge aware conversation generation with explainable reasoning over augmented graphs. 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 1782–1792, 2019.

[23] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 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, 2019.

[24] Shachi Paul, Rahul Goel, and Dilek Hakkani-Tür. Towards universal dialogue act tagging for task-oriented dialogues. *Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech 2019)*, pages 1453–1457, 2019.

[25] Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2182–2192, 2018.

[26] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[27] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, 2002.

[28] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 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, 2018.

[29] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373, 2000.

[30] Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 147–157, 2017.

[31] Yueming Sun and Yi Zhang. Conversational recommender system. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 235–244, 2018.

[32] Ryuichi Takanobu, Runze Liang, and Minlie Huang. 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, 2020.

[33] Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 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, 2019.

[34] Xiaoli Tang, Tengyun Wang, Haizhi Yang, and Hengjie Song. Akupm: Attention-enhanced knowledge-aware user preference model for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1891–1899, 2019.

[35] Dai Hoang Tran, Quan Z Sheng, Wei Emma Zhang, Salma Abdalla Hamad, Munazza Zaib, Nguyen H Tran, Lina Yao, and Nguyen Lu Dang Khoa. Deep conversational recommender systems: A new frontier for goal-oriented dialogue systems. *arXiv preprint arXiv:2004.13245*, 2020.

[36] David R Traum. Speech acts for dialogue agents. In *Foundations of rational agency*, pages 169–201. Springer, 1999.

[37] Yi-Lin Tuan, Yun-Nung Chen, and Hung-yi Lee. Dykgchat: Benchmarking dialogue generation grounding on dynamic knowledge graphs. 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 1855–1865, 2019.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[40] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, pages 592–600, 2018.

[41] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. Knowledge graph convolutional networks for recommender systems. In *The World Wide Web Conference*, pages 3307–3313, 2019.

[42] Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804, 2019.

[43] Yikun Xian, Zuohui Fu, S Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 285–294, 2019.

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

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

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

[47] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 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, 2018.

[48] Tiancheng Zhao and Maxine Eskenazi. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–10, 2016.

[49] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 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, 2020.

[50] Jie Zou, Yifan Chen, and Evangelos Kanoulas. Towards question-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 881–890, 2020.