Explicit Knowledge Graph Reasoning for Conversational Recommendation

XUHUI REN and TONG CHEN, The University of Queensland, Australia
QUOC VIET HUNG NGUYEN, Griffith University, Australia
LIZHEN CUI, Shandong University, China
ZI HUANG and HONGZHI YIN, The University of Queensland, Australia

Traditional recommender systems estimate user preference on items purely based on historical interaction records, thus failing to capture fine-grained yet dynamic user interests and letting users receive recommendation only passively. Recent conversational recommender systems (CRSs) tackle those limitations by enabling recommender systems to interact with the user to obtain her/his current preference through a sequence of clarifying questions. Recently, there has been a rise of using knowledge graphs (KGs) for CRSs, where the core motivation is to incorporate the abundant side information carried by a KG into both the recommendation and conversation processes. However, existing KG-based CRSs are subject to two defects: (1) there is a semantic gap between the learned representations of utterances and KG entities, hindering the retrieval of relevant KG information; (2) the reasoning over KG is mostly performed with the implicitly learned user interests, overlooking the explicit signals from the entities actually mentioned in the conversation.

To address these drawbacks, we propose a new CRS framework, namely, the Knowledge Enhanced Conversational Reasoning (KECR) model. As a user can reflect her/his preferences via both attribute- and item-level expressions, KECR jointly embeds the structured knowledge from two levels in the KG. A mutual information maximization constraint is further proposed for semantic alignment between the embedding spaces of utterances and KG entities. Meanwhile, KECR utilizes the connectivity within the KG to conduct explicit reasoning of the user demand, making the model less dependent on the user's feedback to clarifying questions. As such, the semantic alignment and explicit KG reasoning can jointly facilitate accurate recommendation and quality dialogue generation. By comparing with strong baselines on two real-world datasets, we demonstrate that KECR obtains state-of-the-art recommendation effectiveness, as well as competitive dialogue generation performance.

CCS Concepts: • **Information systems** → **Recommender systems;**

Additional Key Words and Phrases: Conversational recommendation, knowledge graph, preference mining

This work is supported by the Australian Research Council under the streams of Future Fellowship (Grant No. FT210100624), Discovery Early Career Researcher Award (Grants No. DE230101033 and No. DE200101465), Discovery Project (Grants No. DP190101985, No. DP230101196, No. DP240101108, and No. DP240101814), and Industrial Transformation Training Centre (Grant No. IC200100022).

L. Cui and H. Yin are the corresponding authors.

Authors' addresses: X. Ren, T. Chen, Z. Huang, and H. Yin (Corresponding authors), The University of Queensland, Brisbane, Queensland, Australia, 4072; e-mails: x.ren@uq.edu.au, tong.chen@uq.edu.au, helen.huang@uq.edu.au, h.yin1@uq.edu.au; Q. V. H. Nguyen, Griffith University, Gold Coast, Queensland, Australia, 4215; e-mail: henry.nguyen@griffith.edu.au; L. Cui (Corresponding authors), Shandong University, Jinan, Shandong, China, 250100; e-mail: clz@sdu.edu.cn.

2157-6904/2024/07-ART86 $15.00
https://doi.org/10.1145/3637216

ACM Reference format:
Xuhui Ren, Tong Chen, Quoc Viet Hung Nguyen, Lizhen Cui, Zi Huang, and Hongzhi Yin. 2024. Explicit Knowledge Graph Reasoning for Conversational Recommendation. *ACM Trans. Intell. Syst. Technol.* 15, 4, Article 86 (July 2024), 21 pages.
https://doi.org/10.1145/3637216

1 INTRODUCTION

Recently, **conversational recommender systems (CRSs)** have attracted widespread attention from both academia and industry for providing persuasive and explainable recommendations through human-like conversations with users [25, 30, 38, 58]. Different from conventional recommender systems that produce recommendations by analyzing the user preference from their historical interactions, CRSs proactively ask questions to help users explicitly clarify their up-to-date demand during the conversation, achieving dynamic and explainable recommendations.

The general framework for a CRS consists of a **dialogue component (DC)** and a **recommender component (RC)** [14]. The responsibility of the DC is to generate clarifying questions and replies to further the conversation. The clarifying questions guide a user to express preference on the specific attribute types in a **"System Ask–User Answer" (SAUA)** manner [65]. RC extracts the contextual information inside the conversation utterances and recommends suitable items to satisfy the user's preferences when the confidence is high. Recently, a variety of CRSs have been proposed to improve interactions with users and pursue high-quality recommendation [25, 30, 38, 58, 65, 69].

Despite that existing CRSs have demonstrated improvements over conventional recommender systems, two intrinsic issues of CRSs have been hindering their practicality. First, current CRSs heavily rely on the user's responses to the clarifying questions to acquire timely attribute-level demands. However, the considerably large attribute pool in real-world applications leads to numerous system attempts on formulating clarifying questions, which weakens the user experience [28] as a user will be required to answer a long list of questions before receiving the final recommendation [37]. Second, such an SAUA interaction scheme easily triggers a disconnection between the explicit attribute-level and item-level user preferences [69]. While it is common for a user to express her/his preference at both levels, the SAUA mode can only capture the superficial keywords appearing in the utterances, lacking complementary contextual information for fusing attribute- and item-level expressions to better understand user interests. As shown in Table 1, a user is asking for scary movies like "Annabelle," where her/his demand is expressed by attribute-level expression "horror" and item-level expression "Annabelle." Ideally, if the CRS could make full use of utterance contexts, then it may naturally deduce a persuading recommendation with a reasonable explanation, e.g., "a horror film and a look into the early works of James Wan" in the example. However, the straightforward keyword extraction in contemporary CRSs cannot support the comprehension of underlying semantics that bridge "horror" and "Annabelle," not to mention fusing them to generate accurate recommendations. Therefore, a more effective solution for understanding user utterances is needed.

Some recent CRSs incorporate auxiliary **knowledge graphs (KGs)** as complementary resources for analyzing item- and attribute-level user preferences [4, 13, 39, 64]. The rich attribute information carried by KGs are well suited to the need for answering item-specific queries from users and handling cold-start situations. In fact, the inherent graph structure of the KG could naturally link the item- and attribute-level preferences to facilitate explicit reasoning of the user demand [4, 20, 61], providing a more efficient way to engage the DC and the RC. From the perspective of

Table 1. Example of a Conversation for Movie Recommendation

User:	Hi, I am looking for a movie recommendation.
System:	What kind of movies do you like?
User:	I love horror movies similar to Annabelle. I never knew a doll could be so scary.
System:	Dead Silence might be suitable for you! It is a horror film and a look into the early works of James Wan.
User:	Really? I would like to have a try!

Mentioned items and attributes are marked in blue and red, respectively.

the DC, the reasoning process could enable smoother response generation with stronger rationality. The dialogue policy can follow the reasoning flow to proactively explore user interests, chat by referring to informative entities, and provide recommendations with faithful explanations [32]. From the perspective of the RC, reasoning over the KG uncovers the connections between users and items, thus helping to fully comprehend a user's preferences. The extra connectivity information presents chains of evidence to support explainable recommendation, and is also beneficial for tackling the cold-start and data sparsity problems during recommendation [6, 40, 53]. By reasoning over KGs based on the dialogue context, the conversational recommender will be able to provide more accurate recommendations in the early stage of the conversation, and to generate explanations for the item recommended.

In general, existing KG-based conversational recommenders mostly perform reasoning with the implicit information (i.e., representations/embeddings) learned from users' dialogues, especially with the advances in graph neural networks [51, 63]. However, this voids a KG's advantage in explicit reasoning, as the explicitly mentioned entities are largely neglected and all the subsequent reasoning is conditioned on the noisy embeddings of all user utterances, leading to suboptimal KG inference and hence unsatisfying recommendation performance. Furthermore, when performing similarity-based reasoning by comparing user response and KG entities, existing KG-based methods largely neglect the fact that those two representations are learned from two distinct embedding space. As a result, the reasoning performance is further hindered by such semantic discrepancies.

In this article, we propose a **Knowledge Enhanced Conversational Reasoning (KECR)** approach for CRS that conducts explicit reasoning over the KG. The key hypothesis of this work is that an explicit reasoning process on the KG can better stimulate the strengths of a CRS, making it rational and explainable. An accurate and rational graph reasoning process usually requires to learn the entity embeddings and the context embeddings. Therefore, we incorporate an item-based KG (i.e., DBpedia [1]) to model the semantic embeddings of the item/attribute entities, and a pre-trained language model to learn the context embeddings. As these two embeddings are from two different embedding spaces, we additionally utilize mutual information estimation [15]. Thus, we address the challenge of **semantic discrepancy between entity and utterance embeddings** by forcing the related embeddings in two embedding spaces to be closer given the utterance-entity co-occurrences in the conversations. The optimized embeddings encode comprehensive context information and user preferences in a recommendation setting, and constitute the inputs of the reasoning module. To tackle the challenge of **explicit reasoning for retrieving relevant KG entities**, the reasoning module starts from identified entity nodes within KG, and makes entity selection to find a prominent KG path from the mentioned items and/or attributes to a suitable recommendation, constructing a reasoning flow to guide the system to generate responses. Supervised by this explicit graph reasoning fashion, KECR selects the most relevant entities in the reasoning flow to generate human-like responses with a pre-trained language model, which could further guarantee the coherence of the conversation [54].

To summarize, our contributions are threefold:

- We propose a new CRS, namely, KECR that makes full use of the explicit knowledge within KGs to facilitate accurate reasoning and hence high-quality recommendations.
- We point out the bottleneck brought by semantic discrepancies between KG and utterance embeddings, and propose a mutual information maximization constraint in KECR that aligns the semantics between the two embedding spaces.
- We conduct extensive experiments on two real-world datasets to verify the effectiveness of KECR regarding the quality of both the recommendations and generated conversations.

2 RELATED WORK

2.1 Knowledge Graph Reasoning

In recommendation tasks, the key is to accurately capture a user's interest, which is usually multifaceted [47, 48] and dynamic [5, 7]. Furthermore, the cold-start scenario [50] in recommendation makes it extremely challenging to precisely associate users with their preferred items. Performing explicit reasoning on a KG has been demonstrated as an effective solution to retrieving a user's desired items or answers [12, 34, 67]. Prior efforts are focused on multi-hop reasoning over KGs to support question-answering tasks. The queries are augmented with entity graphs or concept nets to enhance their semantic representations, which are then fed into the database to facilitate similarity-based retrieval [44, 57]. Motivated by the superior performance on item prediction, some recommender systems leverage KG embedding techniques to improve item embeddings during the recommendation [4, 37, 60, 69]. However, such methods cannot explicitly utilize the connection information within the graph, resulting in the loss of explainability.

To address this problem, several path-based methods are proposed. The path-based methods are designed to explore prominent paths between users and recommended items with the connection of entities in the knowledge graph [17, 70]. However, they lack an interactive approach to obtain users' needs, and their reasoning processes are limited to pre-defined meta paths and cannot be dynamically refined with users' new inputs [53]. Consequently, their performance heavily relies on the additional domain knowledge for meta-path engineering, which is rather labour-intensive.

2.2 Conversational Recommendation

CRSs have become an appealing approach to bridge the interactions between users and systems for meeting users' real-time demand with explainable and accurate recommendations. This emerging research topic originates from task-oriented dialogue systems that help users complete recommendation-related tasks in a multi-turn conversation [37, 45]. The conventional design of CRSs follows the SAUA fashion [65], where the DC asks a series of clarifying questions over pre-defined attributes to collect the user's explicit preference and use a retrieval/**collaborative filtering-based (CF)** recommendation method [18] to deduce personalized recommendations.

The main limitation of such methods is that, real-world application scenarios contain a large number of candidate attributes that can be used to formulate clarifying questions. As a result, a user may be required to answer many clarifying questions before reaching the final recommendation, which greatly harms the user experience. Also, the dialogue behaviours and content in these methods are strictly restricted. The user and the CRS can only interact with the user's preference on a fixed set of attributes, and the CRS straightforwardly presents recommendations without any explanations [31]. It thus leads to the semantic gap in the utterance understanding, especially when the user reflects his/her preference toward both items and attributes.

To counteract such limitations, some recent approaches adopt KGs to enrich the conversation information to bridge the semantic gap during the understanding [4, 13, 20, 24, 39, 54]. For

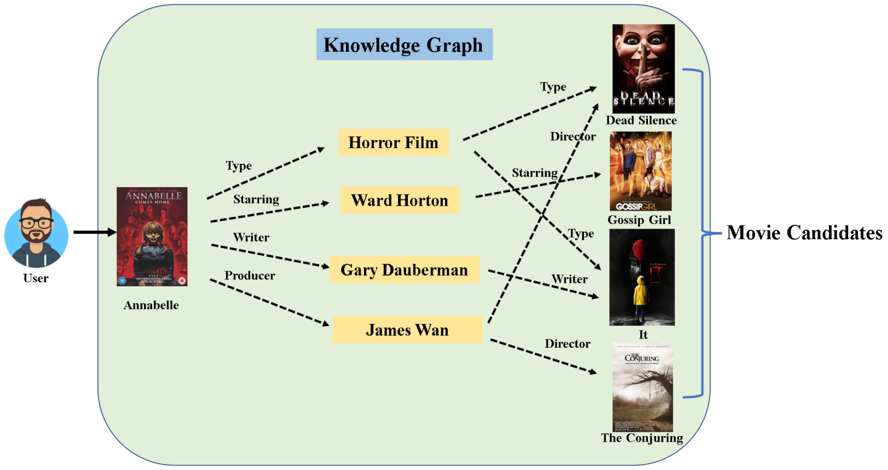

Fig. 1. Illustration of connection information inside the KG in the movie domain. The solid lines between entities represent their relations, while the underlined entity marked in red represents what the user mentioned during the conversation.

example, an item attribute graph is leveraged in References [46, 68] to effectively bind the hierarchical information with the recommendation and conversation generation process. As discussed in Section 1, most existing works along this line do not make full use of the advantages of the knowledge graph (e.g., learning an implicit representation of the subgraph retrieved from the KG [64]). We argue the connectivity information over the knowledge graph is a crucial piece of information for comprehending users' preferences, and explicitly reasoning over the knowledge graph would better carry forward the advantages of CRSs. Furthermore, the semantic discrepancies between KG and utterance embeddings are usually overlooked, which impairs the KG reasoning performance and leads to suboptimal recommendation and dialogue generation. To this end, we devise a novel conversational recommender system by explicitly reasoning the user preference on the knowledge graph, and aligns both the KG and conversation embedding spaces via a mutual information maximization constraint. As such, our model is able to efficiently and precisely learn user preferences and provide reasonable semantic guidance for response generation.

3 PRELIMINARIES

Conversational recommendation is currently an emerging research topic that has aroused exploration in various perspectives [24, 27, 37, 45]. This work follows the basic settings of CRSs [45] but makes some changes to improve its real-world applicability. In a conversation, the CRS is free to interact with the user via several coherent utterances to acquire the user's real-time preferences. According to the collected preferences, the system will deduce suitable recommendations to satisfy the user. The entire conversation ends until the recommendation has been accepted or the user leaves. This article points out that reasoning on the KG better takes advantage of the graph's structural information to generate rational and explainable recommendations. Following the movie recommendation example in Table 1, a user wants a *horror* movie that is similar to *Annabelle* she/he has watched. As shown in Figure 1, such contextual information helps to generate a reasoning path on the knowledge graph to the potential recommendation *Dead Silence* as well as deduce the rationale that the user may prefer this recommendation due to the directing style of *James Wan*. The reasoning process unveils the reason behind a recommended item that distinguishes it from other candidates. So, we incorporate a reasoning function into the CRS setting to endow it with

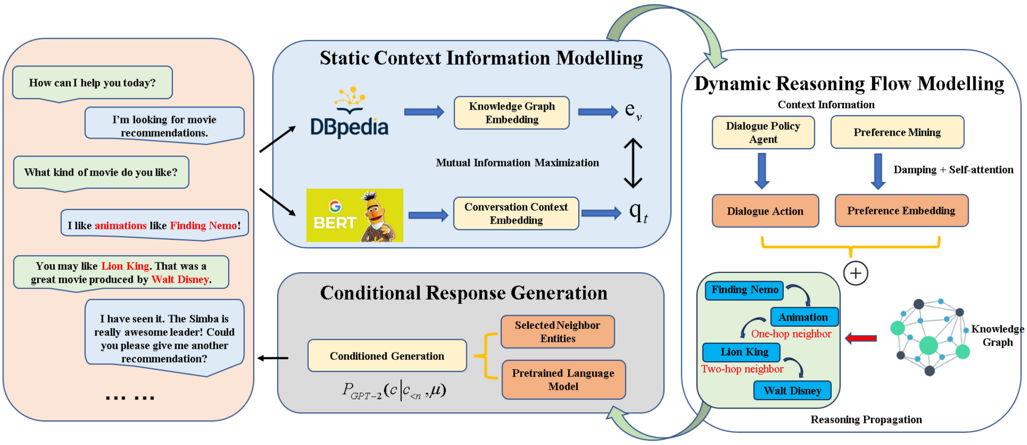
How can I help you today?
I'm looking for movie recommendations.
What kind of movie do you like?
I like animations like Finding Nemo!
You may like Lion King. That was a great movie produced by Walt Disney.
I have seen it. The Simba is really awesome leader! Could you please give me another recommendation?
… …
Static Context Information Modelling
DBpedia
Knowledge Graph Embedding
e_v
Mutual Information Maximization
BERT
Conversation Context Embedding
q_t
Conditional Response Generation
Conditioned Generation
Selected Neighbor Entities
Pretrained Language Model
$P_{GPT-2}(c|c_{<n},\mu)$
Dynamic Reasoning Flow Modelling
Context Information
Dialogue Policy Agent
Preference Mining
Damping + Self-attention
Dialogue Action
Preference Embedding
Finding Nemo
Animation
One-hop neighbor
Lion King
Two-hop neighbor
Walt Disney
Knowledge Graph
Reasoning Propagation

Fig. 2. Overall workflow of KECR. KECR is composed of three key components: Static Context Information Modelling, Dynamic Reasoning Flow Modelling, and Conditional Response Generation. Note that the reasoning propagation illustrated involves only two hops.

explainability. Simultaneously, the reasoning path can be utilized by the DC to generate sensible persuasions, making the system dialogues more appealing and acceptable.

Without loss of generality, we denote a conversation consisting of several interaction rounds by $\mathcal{X} = \{x_t\}_{t=1}^{\phi}$, in which each interaction round x_t refers to a pair of response utterances between the system and a user and ϕ is the number of conversation rounds. To facilitate the reasoning process, we resort to an auxiliary knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, where $\mathcal{V}$ denotes the entity set and $\mathcal{R}$ denotes the relation set. Both items and their associated attributes are represented as nodes (a.k.a. entities) in the KG. Each item entity is linked with its corresponding attribute entities via bidirectional relations, e.g., $Annabelle \overset{Type}{\Longleftrightarrow} Horror\ Film$. In each conversation round t, the system selects a subset of neighbourhood entities $v \in \mathcal{V}$ to enhance the response generation, which could be regarded as a reasoning propagation process over the KG. The purpose is to optimize the reasoning path to maintain the engagement and semantic rationality of the conversation, and output the response utterances and eventually make an item recommendation to the user.

4 METHODOLOGY

We describe our proposed KECR approach in this section. As shown in Figure 2, KECR mainly consists of three components: (1) Static Context Information Modelling that is designed for embedding the static utterances and the KG into latent representations; (2) Dynamic Reasoning Flow Modelling that extracts the instant context information to estimate the user preference and predict the next response action, and then reason over the KG to select proper entities for the downstream task; and (3) Conditional Response Generation that produces human-like responses based on the selected entities. The design of each component is detailed in the following subsections.

4.1 Static Context Information Modelling

As illustrated in Section 1, the diversified expressions of user preferences (i.e., item- and attribute-level expressions) make it difficult to understand the underlying interests within a user utterance, hindering the CRS from deducing accurate recommendations and suitable dialogue responses. We propose to tackle this problem from two perspectives in this article to improve the learning of such context information. First, we introduce a KG to enrich the semantics of a user's expression, thus uncovering the associations between item- and attribute-level user preferences. Then, a pre-trained language model is adopted to learn the semantic representation of the conversation context.

4.1.1 Knowledge Graph Embedding. To build a KG for our CRS, we follow References [4, 69] to reconstruct a subgraph from DBpedia that only contains item entities appearing in our corpus. The original DBpedia only contains directional links from items to their corresponding attributes, which is insufficient for performing embedding and reasoning over the KG in the recommendation context. Therefore, we expand the item-attribute relation to be bidirectional, and add new links from generic categories (e.g., movie genre and release date) to their values, compensating the information within the KG.

As the connected neighbours of an entity could reflect its inherent properties, recent methods based on **graph convolutional networks (GCNs)** have shown state-of-the-art effectiveness in aggregating neighborhood data to enhance the learned entity embeddings of a KG [52, 56, 66]. For modelling this relational information, we adopt R-GCN [41], which generalizes GCN with relation-specific weight matrices, to learn entity representations in a finer granularity. At the $(l{+}1)$th layer, the embedding of each entity $v \in \mathcal{V}$ is learned by propagating information from its one-hop neighbours:

$$\mathbf{e}_v^{l+1} = \sigma\left(\sum_{r \in R} \sum_{v' \in \mathcal{N}_r(v)} \frac{1}{z_v^r} \mathbf{W}_r^l \mathbf{e}_{v'}^l + \mathbf{W}_0^l \mathbf{e}_v^l\right), \tag{1}$$

where σ is the sigmoid function, $\mathcal{N}_r(v)$ denotes the neighbor entity set of v under relation $r \in \mathcal{R}$, z_v^r is the normalization factor, $\mathbf{W}_r^l$ and $\mathbf{W}_0^l$ are learnable weight matrices for integrating the relation-specific neighbors and self-loop at the lth layer, respectively. For notation convenience, we denote the entity embedding at the final layer as $\mathbf{e}_v$, which is used for subsequent computations.

4.1.2 Conversation Context Embedding. During a conversational recommendation process, the system interacts with the user to acquire necessary information about the real-time user preference from the conversation context and generate high-quality recommendations. The learning of the context embedding influences the a CRS's comprehension of the user preference. Recent efforts in adopting a large pre-trained language model (e.g., BERT [11] and GPT-2 [2]) in an open-domain dialogue system have achieved substantial improvements on modelling the conversational context [62], showcasing the potential of exploiting large pre-trained language models in conversational recommendation.

In this vein, we adopt BERT, a pre-trained multi-layer transformer to embed the conversation utterances. In each conversation round, the concatenation of the conversation utterances between the system and the user in $x_t \in \mathcal{X}$ is fed into BERT (as a sequence of word IDs) to obtain its latent embedding $\mathbf{x}_t$. To enhance the contextual connections with the conversation history, we build a **gated recurrent unit (GRU)** [9] to account for long-term sequential dependencies. The context embedding in round t is obtained via

$$\mathbf{q}_t = GRU(\mathbf{q}_{t-1}, \mathbf{x}_t). \tag{2}$$

In addition to the context embedding $\mathbf{q}_t$, the entities mentioned in the conversation are in particular important for understanding the user's preference. To recognize these entities during the conversation, we perform fuzzy matching with the lemmatized form of user utterances using SpaCy[1] and map them into the KG. It is worth noting that, the main innovation of our work lies in how

[1] https://spacy.io/

we perform context-aware KG reasoning to enhance the conversational recommendation performance, hence we adopt the widely used Spacy as an off-the-shelf tool for entity extraction. In addition, Spacy can provide sufficient accuracy for most recommendation domains (e.g., movies) to facilitate subsequent tasks. In round t, we store the KG embeddings of entities in all previous t rounds as a dialogue belief state, denoted by $\mathbf{D}_t = [\mathbf{e}_{v_1}, \mathbf{e}_{v_2}, \cdots, \mathbf{e}_{v_E}]$ with E chronologically sorted entities.

4.1.3 Learning Embeddings Via Mutual Information Estimation. We hereby describe how we jointly learn both KG and conversation context embeddings. Given the constructed KG and full conversation $\mathcal{X}$, one may choose to learn each type of the embeddings separately. However, this aggravates semantic discrepancies as the representations of the KG entities and conversation context are generated from two distinct embedding spaces. In an utterance, as the mentioned entities v are informative and heavily entangled with the conversation context, we take advantage of the mutual information (MI) maximization trick [43, 59, 69] to align and fuse highly relevant KG and conversation embeddings. The main idea is to calculate the MI between two coupled variables, i.e., the co-occurring entity $v \in \mathcal{V}$ and utterance $x_t \in \mathcal{X}$ in our case:

$$MI(v, x_t) = D_{KL}(p(v, x_t) \| p(v)p(x_t)), \tag{3}$$

where D_{KL} is the Kullback-Leibler divergence between the joint distribution $p(v, x_t)$ and product of marginals $p(v)p(x_t)$.

Given the intractable computation of the precise MI in practice, we adopt a neural approach to estimate MI using Jensen-Shannon divergence [15], which could be implemented in the following binary cross-entropy (BCE) form L_{MI}:

$$\begin{aligned} L_{MI} &= \mathbb{E}_P[\log(g(\mathbf{e}_{v^+}, \mathbf{q}_t))] + \mathbb{E}_N[\log(1 - g(\mathbf{e}_{v^-}, \mathbf{q}_t))] \\ &\leq MI(v, x_t), \end{aligned} \tag{4}$$

where $\mathbb{E}_P$ and $\mathbb{E}_N$ denote the expectation over all positive and negative samples, respectively. $g(\cdot, \cdot)$ is a neural classifier, which inputs the concatenation of the embedding vectors of one entity and utterance and computes a probability in the range $[0, 1]$. In KECR, for every positive sample (v^+, x_t) consisting of a co-occurring utterance and a mentioned entity, we pair it with four negative samples (v^-, x_t) by randomly drawing entities that do not appear in utterance x_t.

By maximizing L_{MI}, we can essentially push up the lower bound of MI. In this static context information modelling process, it helps us effectively pre-train both the KG embeddings and the GRU encoder in Equation (2) by calibrating and fusing their mutual information. This brings more expressiveness to $\mathbf{e}_v$ and $\mathbf{q}_t$ generated, which will be used and further finetuned for subsequent reasoning and recommendation tasks.

4.2 Dynamic Reasoning Flow Modelling

Conditioned on the static information extracted from the KG and utterance, we need to dynamically comprehend the real-time user preference and reason over the KG to find proper entities to push forward the conversation and/or generate an accurate recommendation. There are mainly two factors that influence the reasoning process, namely, the dialogue action the model is about to execute (e.g., query or recommend) and the user preference mined from the conversation. The generated dialogue action will help rectify the reasoning flow to be coherent with the conversation context. Meanwhile, the user preference will help select entities that are the most suitable for the user's demand. In this section, we will detail how both factors are computed and then utilized for explicit reasoning over the KG.

4.2.1 Dialogue Policy Agent. The dialogue policy agent decides which dialogue action should be performed conditioned on the current conversation context. Previous CRSs usually define the dialogue action from the perspective of the SAUA-based interaction scheme, where each action is defined as one yes/no question about a specific attribute, e.g., "Do you like *action* movies?" As pointed out recently in References [23, 37], this leads to an unnecessarily large action space for training a performant policy agent, and is less efficient in acquiring user preferences. In this regard, we narrow down the action space to only three types of actions: query, recommend, and chit-chat, while the reasoning process further decide which relevant attribute (i.e., the KG entity) will be incorporated for composing the conversational response.

Inspired by prior works on task-oriented dialogue systems that estimate system actions based on the summarization of the conversation context [31, 55], we train a policy agent with a multi-layer perceptron classifier to decide the dialogue action in each conversation round. As the pre-trained conversation context encoder is sufficiently expressive, we only take the conversation embedding as the input to the policy agent:

$$\mathbf{a}_t = \text{softmax}(\mathbf{W}_1\text{ReLU}(\mathbf{W}_2\mathbf{q}_t + b_2) + b_1), \tag{5}$$

where $\mathbf{a}_t$ represents the probability distribution over three dialogue actions, $\mathbf{W}_1, \mathbf{W}_2, b_1, b_2$ parameterize the policy agent with learnable weights and biases. The policy agent is learned by minimizing the following cross-entropy loss:

$$L_a = -\sum_{\forall i} a^*_{t,i} \log(a_{t,i}), \tag{6}$$

where $a^*_{t,i}$ and $a_{t,i}$ denote the ith element of the ground truth and prediction, respectively.

4.2.2 Preference Mining. As introduced in Section 4.1.2, in each round, the informative entities mentioned in the conversation are stored in $\mathbf{D}_t$. Intuitively, the encodings of these informative entities are strong reflections of the user preference. Instead of simply averaging these vectors, we further consider two marginal effects from the context and time perspectives.

The context-awareness requires identifying the dominant features among the input context rather than treating all entities equally. Therefore, we adopt a simplified self-attention mechanism [29] to calculate the context-level user preference representation:

$$\mathbf{u}_{cont} = \mathbf{D}_t \cdot \text{softmax}(\mathbf{W}_3\tanh(\mathbf{W}_4\mathbf{D}_t)) \tag{7}$$

where $\mathbf{W}_3$ and $\mathbf{W}_4$ are weight matrices.

The time marginal effect comes from the topic/interest shifts during the conversation. The user may change his/her mind and request for different recommendations within a conversation. Therefore, we adopt a damping factor $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_E]$ to control the intensity of the information influence from the conversation history, and the time-level user preference is formulated as

$$\mathbf{u}_{time} = \mathbf{D}_t\boldsymbol{\gamma}. \tag{8}$$

With that, the final user preference embedding is the mean between $\mathbf{u}_{cont}$ and $\mathbf{u}_{time}$:

$$\mathbf{u}_t = \frac{\mathbf{u}_{cont} + \mathbf{u}_{time}}{2}. \tag{9}$$

4.2.3 Heterogeneous Reasoning over the KG. To perform explicit reasoning on the KG, we devise a heterogeneous graph reasoning module to expand the dialogue action and user preference into a reasoning path to select the right entity for dialogue generation, e.g., choosing an attribute to query about or identifying items to recommend.

Specifically, the starting point of a reasoning process is the entity v_{t-1} appearing in the previous round of the conversation. If no entity has been mentioned in the user utterance at the early stages of the conversation (e.g., $t = 1$), then one generic attribute type (e.g., actor, type, genre in movie domain) will be randomly selected to formulate a clarifying question. As the utterances in conversational recommendation tend to be short, few utterances contain multiple entities. If the user does mention multiple entities in a single utterance, then one will be randomly selected as the starting point. Though selecting the entity with higher relevance is more desired on such occasions, it requires either the user profile or interaction history to support semantic matching. However, our task setting of conversational recommendation is subsumed under a strict cold-start scenario. Specifically, each complete multi-turn dialogue is treated as an individual case generated by an anonymous user (i.e., no prior knowledge about the user is assumed). As per the design of KECR, its reasoning process needs to leverage the explicit entity v_{t-1} mentioned in the previous dialogue turn, which is infeasible at the beginning when $t = 1$. To cope with this setting, KECR selects one generic attribute type (e.g., actor, type, genre in movie domain) uniformly at random. In this way, KECR can effectively formulate a clarifying question for the first conversation turn without requesting for additional user profile information. As the conversation stretches to multiple turns, KECR will be able to gradually refine its KG reasoning based on the relevant entities identified from earlier user response.

With the starting point v_{t-1}, the reasoning target is to determine the next entity of the reasoning path from its neighbors $v \in \mathcal{N}(v_{t-1})$, and the next entity should be coherent to the given contexts. Therefore, we need to calculate a relevance score to evaluate the confidence on each neighbour node under the current context, namely, the user preference embedding $\mathbf{u}_t$, conversation context $\mathbf{q}_t$, and dialogue response action $\mathbf{a}_t$:

$$\begin{aligned}\mathcal{J}(k) &= \sigma(\mathbf{h}_k \mathbf{W}_{proj} \mathbf{h}_c),\\ \mathbf{h}_k &= [\mathbf{e}_{v_{t-1}}; \mathbf{e}_k],\\ \mathbf{h}_c &= [\mathbf{a}_t; \mathbf{u}_t; \mathbf{q}_t],\end{aligned} \tag{10}$$

where k indexes entity $v_k \in \mathcal{N}(v_{t-1})$, and $\mathbf{e}_v$ and $\mathbf{e}_k$ are, respectively, the KG embedding of the starting and neighbour entities. The relevance score $\mathcal{J}(k)$ denotes each v_k's relevance to the starting point v_{t-1}. The neighbour that holds the highest relevance score will be selected in the reasoning path. Since this neighbour selection process could be regarded as a binary decision on each $v_k \in \mathcal{N}(v_{t-1})$, we optimize this process by minimizing the following BCE loss:

$$L_r^j = -\sum_{v_k \in \mathcal{N}(v_{t-1})} \mathcal{J}(k) \log \mathcal{J}(k) + (1 - \mathcal{J}(k)) \log(1 - \mathcal{J}(k)), \tag{11}$$

where the loss is notably indexed by j. The rationale is, when a recommendation action is needed and an item entity is reached as the one-hop neighbor of v_{t-1}, we expect our system to be able to provide both the recommendation and the reason behind this item selection. Therefore, after selecting the one-hop neighbor of v_{t-1}, in KECR we propose to repeat the reasoning process from this selected neighbor entity. The reasoning is performed analogously using Equation (10). As such, it can reason further down the KG path, retrieve attributes closely associated with the item, then formulate an explanation to enhance the rationality and persuasiveness of the system response. At the same time, it also makes the second reasoning step easily impacted by the result of the first step. Hence, we append a small weight λ to the second reasoning step $j = 2$ to avoid unexpected cumulative error. The final reasoning loss L_r is

$$L_r = L_r^1 + \lambda L_r^2. \tag{12}$$

4.3 Conditional Response Generation

The generator integrates selected entities and the context into the decoding process and emits a response, which could be regarded as a query expansion problem in language generation. In view of the prior success of transformer-based language models in expanding a concept to a descriptive sentence [3, 20, 42] and the state-of-the-art performance of pre-trained language generation models, training a language generation model from scratch is unnecessary. It is also worth mentioning that our innovation mainly lies in learning expressive KGs and conversation representations via mutual information maximization as well as the explicit KG reasoning. Thus, we adopt GPT-2 [36] in KECR to generate the response for the next turn $y_{t+1} = \{c_1, c_2, \cdots, c_N\}$ and it consists of N tokens: $c_n = \text{argmax}_{\forall c} P_{\theta_{GPT-2}}(c \mid c_{<n}, \mu)$, where μ denotes the two entities resulted from the reasoning process earlier, and θ_{GPT-2} denotes the pre-trained GPT-2 model that parameterizes the probability calculation of token c. When generating the nth token in the response, GPT-2 encodes the concatenated embeddings of selected entities μ and the context x_t, then decode tokens inside the response sequentially with nucleus sampling method [2, 16], which helps to generate semantically richer responses.

4.4 Training

The training process of our proposed KECR contains two phases: (1) pre-training for the static context information modelling; and (2) joint training for the dialogue policy agent and dynamic reasoning module. For the first phase, the training objective is to fuse the mutual information from the graph embedding and the context embedding via mutual information estimation. We optimize the semantic space of each type of embedding by maximizing the lower bound of their mutual information defined by the BCE loss L_{MI} in Section 4.1.3. For the joint training phase, we first infer the response action based on the conversation context. Then, we integrate the damping effect and the self-attention mechanism to derive the user preference embedding. Conditioned on the dialogue action, context embedding, and user preference embedding, KECR reasons on the KG to find proper neighbours to formulate the next response. KECR's model parameters are refined by minimizing the sum of L_a and L_r. Note that GPT-2 has already been trained and optimized on a massive Web corpus, hence we directly inherit the original GPT-2 model to decode the conversational response. The detailed training process can be found in Algorithm 1.

5 EXPERIMENTS

We evaluate KECR on two real-world datasets. Specifically, we aim to answer the following **research questions (RQs)**:

RQ1: How does our proposed KECR perform compared with state-of-the-art conversational recommendation methods?

RQ2: Are our mutual information estimation and preference reasoning modules really effective?

RQ3: How can KECR smoothen the conversation flow and provide persuading recommendations?

5.1 Datasets

We conduct experiments on two publicly available datasets, REDIAL [25] and GoRecDial [21], which were collected with Wizard-of-Oz paradigm [22]. The conversations in both datasets are originally generated by human volunteers recruited via Amazon Mechanical Turk, a popular crowdsourcing site [19]. The raw corpus contains many grammatical errors and incomplete tokens, so we manually corrected them before the training. Both datasets focus on the movie recommendation domain. REDIAL involves the free chats between the wizard and the seeker. The wizard captures the user preference and selects a suitable recommendation for the seeker. At least

Table 2. Statistics of REDIAL and GoRecDial

		REDIAL	GoRecDial
Conversations	Interactions	10,006	9,125
	#Utterances	182,150	170,904
	#Movies	6,924	3,782
KG	#Entity	30,471	19,308
	#Relation	12	12
	#Triple	392,682	227,384

ALGORITHM 1: The training algorithm for KECR.

```
Input: Knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, training data $\mathcal{X}$
Result: Fine tuned parameter for graph embedding $\theta_g$, context embedding $\theta_c$, policy agent $\theta_a$,
        reasoning module $\theta_r$
Randomly initialize $\theta_g$, $\theta_c$, $\theta_a$, $\theta_r$;
Pre-train $\theta_g$ and $\theta_c$ by maximizing the MIM loss $L_{MI}$ in Equation (4);
for $t = 1 \rightarrow |\mathcal{X}|$ do
    Acquire the entity embedding $e_v$ from $\mathcal{G}$ and the context embedding $q$ from $\mathcal{X}$ by Equations (1)
      and (2), respectively;
    Predict the dialogue action $a$ by Equation (5);
    Compute user preference representation $u$ by Equations (7) and (9);
    Perform reasoning over the KG from the starting point, calculate the relevent score of neighbors by
      Equation (10);
    Conduct second step reasoning by Equation (10);
    Update $\theta_a$ and $\theta_r$ with Equations (6), (11), and (12)
end
Return $\theta_g$, $\theta_c$, $\theta_a$, $\theta_r$
```

four different movies are mentioned in a single conversation. GoRecDial additionally provides the watching history of the seeker. The wizard chats with the seeker to acquire the seeker's preference and then recommends a movie from five given candidate movies.

To fully simulate realistic application scenarios, we construct KG by mapping movies to DBPedia via exact name matching, and filter out inappropriate relations. Finally, the relation of *Actor*, *Director*, *Time*, *Genre*, *Subject* are retained from the DBPedia. Apart from these relations, we append the inverse relation in our KG to enhance the graph embedding of the attribute entities, enabling item and attribute entities to have enough neighbours for embedding learning. We also create a *Belong* relation, that connects attributes with general clarifying entities defined by relations. Table 2 illustrates the statistics of the two datasets along with their KGs after preprocessing.

5.2 Experimental Settings

5.2.1 Implementation Details. We implement our CRS with PyTorch. We randomly split each dataset for training, validation and test with the ratio of 8 : 1 : 1. For convenience, the graph and the context embeddings share the same dimension size of 128. The embedding layer of the RGCN L is set to 1, and the normalization factor $Z_{v,r}$ is set to 1. The damping factor is set to 0.95 in Equation (9). We adopt a learning rate of 0.001, and the batch sizes are, respectively, 10 and 30 for pre-training and joint training. Adam is adopted as the optimizer with a weight decay of 0.01. The pre-trained language model BERT and GPT-2 used in KECR are the basic types that are published in Transformers toolkit.[2] KECR did not involve fine-tuning of the pre-trained language model. Therefore, we freeze the parameters of BERT and GPT-2 during the training process.

5.2.2 Baselines. Research on CRS has emerged in recent years, which explores the application in various scenarios and settings. To verify the performance of our proposed KECR, we select several state-of-the-art baseline methods designed for the same application setting as our proposed method. The baselines are summarized as follows:

- ReDial: This is a benchmark model for REDIAL [25]. It consists of an RNN-based language generation module, a CF-based recommender and a sentiment analysis module.
- KBRD: This is a knowledge-based CRS [4] that allows mutual enhancement between the knowledge of user preferences and the generated recommendation results.
- KGSF: This method tackles the semantic gap problem during the conversational recommendation process [69]. Based on the knowledge-enhanced embedding of user preference, the recommender component generates suitable recommendations and feeds them to a transformer-based decoder to generate responses.
- CRWalker: This method transforms the reasoning process of user preference during the conversational recommendation as a tree-based walk on the KG [31]. The model expands the user preference reasoning tree from the root to the end of the leaves. The decoder integrates the reasoning tree to formulate the responses at each conversation turn.

Although there are other recent CRSs [10, 23, 24, 37], they are inapplicable for comparison due to different task settings. For example, [23, 24] are designed to formulate a series of yes/no questions based on the preference estimation, and [10] is designed to learn the dialogue policy during conversational recommendation with reinforcement learning.

5.2.3 Evaluation Metrics. We evaluate the performance of different methods in two ways, automatic evaluation and human evaluation. The automatic evaluation includes the assessment of recommendation accuracy and the quality of language generation. Following Reference [69], we adopt **Recall@k (R@k)** to evaluate the recommendation performance, and **Distinct n-gram (Dist-n)** to measure the diversity of generated responses. Different from the previous methods that measure the performance on sentence-level, we evaluate the performance on context-level, which is more comprehensive for the conversation scenario.

The automatic evaluation metrics sometimes may not reflect the actual property of the generated responses [33]. The similarity-based evaluation metric (i.e., BLEU) may cheat the decoder to generate responses containing more similar tokens, ignoring the grammar and fluency. Therefore, we follow Reference [30] to invite external human volunteers to give their ratings (ranging from 1 to 5) on 4 properties of the generated responses, *fluency*, *coherence*, *informativeness*, and *syntax*.

5.3 Performance Comparison

Table 3 and Table 4 present the experimental results of our proposed method as well as the baselines[3]. Obviously, our proposed method KECR achieves the best recommendation performance and the second-best performance on response diversity.

Recommendation Effectiveness. As shown in Table 3, most KG-based methods achieve better recommendation performance compared with CF-based ReDial. That is mainly because ReDial generates the recommendation solely relies on the collaborative signal in the training corpus. The use

[2] https://github.com/huggingface/transformers

[3] As mentioned earlier, GoRecDial provides five recommendation candidates during the conversation, so we only evaluate R@1 on the GoRecDial.

Table 3. Recommendation Performance Comparison of all Methods on Two Datasets, Where the Best Performance Is Boldfaced

	REDIAL			GoRecDial	
	R@1 (%)	R@10 (%)	Impr. (R@1)	R@1 (%)	Impr. (R@1)
ReDial	1.93	10.30	+102.59%	69.18	+14.21%
KBRD	3.02	15.73	+29.47%	74.52	+6.03%
KGSF	3.78	16.15	+3.44%	77.36	+2.13%
CRWalker	2.05	11.42	+90.73%	72.83	+8.49%
KECR	**3.91**	**16.49**	–	**79.01**	–

Improvements over all baselines are statistically significant with $p < 0.01$.

Table 4. Response Generation Performance Comparison of All Methods on Two Datasets by Distinct-n, Where the Best Performance Is Boldfaced

	REDIAL			GoRecDial		
	Dist-2	Dist-3	Dist-4	Dist-2	Dist-3	Dist-4
ReDial	10.61	13.07	12.40	11.66	13.23	12.61
KBRD	11.38	23.44	30.21	13.46	17.19	24.92
KGSF	13.45	27.53	34.51	14.14	18.66	27.59
CRWalker	**27.24**	**39.61**	**49.57**	**22.63**	**32.36**	**43.12**
KECR	<u>**26.45**</u>	<u>**36.10**</u>	<u>**42.18**</u>	<u>**18.13**</u>	<u>**27.62**</u>	<u>**39.15**</u>

We additionally underline the second-best performance on Distinct-n.

of plain CF makes ReDial unable to incorporate auxiliary information (e.g., KG) to distill the user preference, hindering it from generating accurate recommendations, especially in the sparse conversation scenario. KBRD provides an effective method to incorporate the KG as complementary information to enrich the item embeddings from RGCN. It indicates the effectiveness of using KGs to improve CRSs. KGSF utilizes two KGs, i.e., ConceptNet and DBPedia, to fuse the token-based and item-based semantics, to facilitate fine-grained user preference modeling and recommendation. However, both KG-based methods only focus on enriching the item/token embeddings during the conversation. They cannot utilize the explicit KG connections to reason the user preference and narrow down the recommendation candidates. CRWalker introduces a reasoning module to generate the recommendation, but it ignores the context information that can help optimize the reasoning process, resulting in inferior recommendation performance.

Dialogue Quality. From the perspective of response generation, Table 4 gives a fair comparison of all methods, where ReDial still makes the worst generation performance. ReDial generates the response with the RNN-based encoder-decoder architecture, which strictly follows the language rules learned from the training corpus. The lack of auxiliary information from KG makes it unable to fully distill the semantics from the conversation context, failing to generate high-quality responses. KBRD and KGSF introduce the vocabulary bias from KG into a transformer module to enrich the response representation. However, as mentioned above, their methods mainly focus on enriching the item/token embeddings during the conversation. They cannot utilize the explicit KG connections to justify and explain their recommendations. CRWalker achieves the highest score on Dist-n. However, as we will discuss further, this is mainly because their inaccurate predictions of recommendation will further mislead the subsequent reasoning process and bring irrelevant vocabulary biases to contaminate the decoding process, which leads to abnormally large Dist-n scores.

Performance Summary. Our proposed KECR achieves advantageous performance on both Dist-n and Recall via explicit reasoning on the KG. The main goal for a CRS is to recommend suitable items for users, where KECR achieves the best performance compared with baselines w.r.t. R@K. The reasoning process could identify informative paths on the KG to find suitable recommendations when scarce preference information could be accessed. Beyond the recommendation accuracy, we want the system to generate diverse responses instead of similar responses as in the training corpus. As measured by Dist-n, KECR only falls behind CRWalker, which is also a KG-based method. The main defect of CRWalker in KG inference is that, CRWalker only utilizes the dialogue information in an explicit way. Specifically, its reasoning process forms a tree with the abstract intent representation extracted from the conversation as the root (i.e., starting point). As CRWalker's reasoning is based on a virtual, implicit "node," CRWalker's reasoning path can be theoretically generated from a wider range of KG nodes, leading to a less accurate but more diverse set of retrieved entities. Eventually, this contributes to the higher vocabulary diversity in the dialogue response generated by CRWalker as measured by Dist-n metrics, which actually comes at the cost of sacrificing the recommendation accuracy. In contrast, KECR identifies the exact entity v_{t-1} mentioned in the previous conversation and starts from there, hence can accurately narrow down the candidate entities when performing reasoning for the next step. This advantage translates to the best recommendation performance and high conversation quality of the proposed KECR.

Discussion on Inference Efficiency. To better align with practical cloud service deployment requirements, we conducted latency tests on a single Nvidia A-100-80G card server. This testing involved the assessment of latency across three distinct phases: static context information modeling, dynamic reasoning flow modeling, and conditional response generation, resulting in respective time costs of 0.219s, 0.031s, and 0.113s. The total inference time, utilizing the FP32 (Float32 single precision) model, was effectively managed to remain at approximately 0.363s, ensuring a satisfactory user experience during conversations. This latency time reflects the inherent inference time without any optimization. In real-world application scenarios, cloud service providers typically implement a series of model optimization techniques, such as BF16 (Bfloat16) optimization, model quantization, and flash attention, to continuously enhance performance. These optimization measures can result in a notable acceleration of inference time, often achieving two times or more improvement.

5.4 Ablation Study

The static context information modelling and the dynamic reasoning flow are the two main components of our proposed KECR. We especially introduce the mutual information estimation and dynamic reasoning module in each part to enhance the performance of the basic CRS. To verify the effectiveness of newly proposed components in our system, we design two variants of our KECR, **KECR-M**, and **KECR-R**. KECR-M keeps all other components but removes mutual information estimation. KECR-R maintains the mutual information estimation to fuse the two channel embedding signals but makes changes on the reasoning part. Compared with KECR that uses a similarity-based function to find the next entity in the KG, we let KECR-R to randomly select one entity from the neighbor set of the current node in the reasoning path. This is to testify whether or not the systematically constructed, context-aware reasoning chain in KECR contributes positively to the recommendation. When performing two-step reasoning, we repeat the random selection process twice, where the only difference is the changed neighbor set based on the current entity. To demonstrate the performance variation, we utilize Dist-3 for dialogue quality and R@10/R@1 (on REDIAL/GoRecDial) for recommendation.

From Table 5, we can observe that the default setting of our proposed method still keeps the best recommendation performance over these two variants. The improvement of the score on R@K for

Table 5. Performance Comparison of all Ablation Methods on Two Datasets by Recall and Distinct-3, Where the Best Performance Is Boldfaced

	REDIAL		GoRecDial	
	R@10 (%)	Dist-3	R@1 (%)	Dist-3
KECR	**16.49**	36.10	**79.01**	27.62
KECR-M	15.63	37.18	76.52	28.46
KECR-R	7.36	**45.64**	43.82	**48.6**

default setting represents both the mutual information estimation and the reasoning module could provide a better guideline for finding the target items. KECR-M shows constant performance degeneration compared with the default one, demonstrating the effect of the mutual information estimation. We also notice that KECR-R has achieved a higher score on Dist-3 compared with the default one. That is mainly because the random selection policy introduces false tokens into the final decoding stage, which would increase the similarity-based Dist-3 score. The improvement on Dist-3 score also comes with the performance damping on R@K. KECR-R cannot accurately identify the user preference and recommend suitable items to the user, failing in the primary task of a conversation recommendation. Meanwhile, we notice a slight increase in Dist-3 with KECR-M. The mutual information constraint essentially ensures the semantic alignment between the context embedding $\mathbf{q}_t$ and corresponding entity embedding $\mathbf{e}_v$, which is designed to support precise reasoning on the KG. Once this constraint is removed, KECR-M produces more unexpected entities in the retrieved reasoning path due to the lowered KG reasoning efficacy. As a result, compared with KECR, the relatively higher vocabulary diversity from those unexpected entities leads to a slight increase in Dist-3 metric.

5.5 Human Evaluation

The automatic evaluation metrics sometimes overly rely on the similarity of the n-gram features, which cannot comprehensively evaluate the generated response's actual quality, e.g., fluency, coherence, informativeness and syntax. However, real-world applications care more about the generated sentence's actual quality than various evaluation metrics. Therefore, we follow Reference [30] and recruit ten external volunteers to contribute their ratings (ranging from 1 to 5) on the generated conversations from four perspectives: fluency, coherence, informativeness, and syntax. A higher score means the volunteers have a higher agreement on one specific perspective of language quality. We sampled 20 conversations from REDIAL for our proposed method and different baselines.

Table 6 presents the collected blind reviews from the human volunteers. KGSF, KBRD, and CR-Walker obtain a satisfying score on four evaluation aspects, mainly because of the adoption of the transformer-based language model and the auxiliary information from the KG. ReDial inherits the property of the RNN-based language models that tends to imitate the conversation behaviour appearing in the training corpus when decoding, which could achieve a good performance on fluency and coherence. Finally, our proposed KECR consistently performs better than all baselines. KECR could make reasoning over the KG to provide informative entities as guidances to enhance the coherence and the informativeness of the generated response. Also, it gets benefits from the powerful pre-trained language model, which is trained on massive open web text. The pre-trained language model expands the key queries obtained from the reasoning process to formulate the response, proving the fluency and the syntax of the generated responses.

Table 6. Human Evaluation of All Methods on Sampled Conversation on Fluency, Coherence, Informativenss, and Syntax, Where the Best Performance Is Boldfaced

	Fluency	Coherence	Informativeness	Syntax
ReDial	4.1	4.2	3.2	3.5
KBRD	4.2	4.0	4.1	4.1
KGSF	4.1	4.0	4.2	4.1
CRWalker	4.0	4.3	4.3	4.4
KECR	**4.4**	**4.6**	**4.7**	**4.5**

Table 7. Sample Conversation between Our KECR (Wizard) and a Real User (Seeker)

Role	Chat Content	Response Action	Reasoning
User	Hello, how are you today?	—	—
AI Agent	Good! What kind of movie are you looking for today?	Query	Genre
User	I really enjoy The Martian. Do you know any other movie that is similar?	—	—
AI Agent	I would recommend Arrived, which is an Amercian science fiction film you may like.	Recommend	Amercian science fiction film; Arrival
User	OK, I also like superhero movie.	—	—
AI Agent	My brother says that Deadpool is a good film.	Recommend	Deadpool; 2010
User	Excellent, I think it will be one of my movie for this weekend.	—	—
AI Agent	Good luck. Goodbye.	Chat	None

The AI Agent is supported by our KECR.

5.6 Qualitative Analysis

In this part, we present two qualitative examples to illustrate how KECR is able to improve the quality of generated responses and generate explainable recommendations with persuasive descriptions.

Table 7 presents a conversation between our human volunteer and our proposed system KECR. The human volunteer interacts with the system to ask for movie recommendations for the weekend night. The conversation starts with the greeting from the volunteer to activate the service. KECR first identifies the dialogue action as "Query" and selects "Genre" to ask for the user preference. Then, the user sequentially expressed his/her preference either with the item entity "The Martian" and the attribute entity "Superhero." The dialogue action generated from the policy agent for both conversation turns are "Recommend." KECR starts to make reasoning on the KG to find a proper movie that shares the same property for the recommendation, and a corresponding reason (i.e., the second-step reasoning process). Conditioned on the selected entities, a human-like persuasive response is generated from our GPT-2-based decoder to recommend the item. Finally, there is no newly mentioned entity appearing in the conversation and the user accepts the recommendation. KECR turns to "Chat" action and says "Goodbye" to the user.

There is an interesting response during the conversation, that KECR introduces "My brother" in the third turn of conversation to strengthen the confidence of the recommendation. That is a side benefit from the pretrained language model that makes our system easier to generate human-like responses. This satisfying performance holds significant promise, particularly in the context of the burgeoning live streaming e-commerce sector. Our KECR can empower virtual live streaming hosts

Table 8. Conversation Example of Live Streaming for Movie Recommendation

Role	Chat Content	Response Action	Reasoning
User	Hello!	—	—
AI Agent	Hi! How can I help you today?	Chat	None
User	Good! I want to find a Marvel Comics today to enjoy the time with my friend.	—	—
AI Agent	How about Spyder-Man? It is a grate superhero fiction comics.	Recommend	Superhero fiction; Spider-Man
User	Yeah, I know it! Good recommendation. Thanks a lot	—	—
AI Agent	You are welcome!	Chat	None

to engage with the audience and offer tailored recommendations to individual users. As shown in Table 8, our model could efficiently locate the user-preferred item and recommend the movie with an excuse, ensuring a gratifying interactive experience.

6 CONCLUSIONS AND FUTURE WORK

In this article, we propose a novel knowledge-enhanced conversational reasoning approach for CRS. We first utilize auxiliary structured knowledge from the KG to bridge the semantic gap between the attribute- and item-level representations, and adopt mutual information maximization to align the semantic space of the context and the knowledge graph. Conditioned on the aligned semantic representations, we make explicit reasoning over the KG to find a prominent reasoning chain to enhance the recommendation and the conversation generation performance. Extensive experiments on two real-world datasets demonstrate the superiority of our proposed system over state-of-the-art baselines. One potential limitation of our approach is its use of several computationally-heavy components, e.g., the R-GCN and the GRU for representation learning. With an ongoing trend of computationally efficient and even on-device recommendation [8, 26, 35, 49], a promising direction for our future work is to delve into lightweight CRSs that support efficient on-device inference. As another line of possible future work, we will consider introducing multi-modal external resources (e.g., videos, demographics) to enrich the learned representations of CRS. Apart from this, we will explore the implementation of our system in real-world conversation scenarios with more conversation behaviours (e.g., chit-chat, ticket booking, movie arrangement) to fulfill the diverse interests of a user.

REFERENCES

[1] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia—A crystallization point for the web of data. *J. Web Semant.* 7, 3 (2009), 154–165.

[2] Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2—How can I help you? Towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd NGT*. 15–22.

[3] Jifan Chen, Shih-Ting Lin, and Greg Durrett. 2019. Multi-hop question answering via reasoning chains. Retrieved from https://arXiv:1910.02610

[4] 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 EMNLP-IJCNLP*. 1803–1813.

[5] Tong Chen, Hongzhi Yin, Hongxu Chen, Rui Yan, Quoc Viet Hung Nguyen, and Xue Li. 2019. Air: Attentional intention-aware recommender systems. In *Proceedings of the ICDE*. 304–315.

[6] Tong Chen, Hongzhi Yin, Jing Long, Quoc Viet Hung Nguyen, Yang Wang, and Meng Wang. 2022. Thinking inside the box: Learning hypercube representations for group recommendation. In *Proceedings of the SIGIR*. 1664–1673.

[7] Tong Chen, Hongzhi Yin, Quoc Viet Hung Nguyen, Wen-Chih Peng, Xue Li, and Xiaofang Zhou. 2020. Sequence-aware factorization machines for temporal predictive analytics. In *Proceedings of the ICDE*. 1405–1416.

[8] Tong Chen, Hongzhi Yin, Yujia Zheng, Zi Huang, Yang Wang, and Meng Wang. 2021. Learning elastic embeddings for customizing on-device recommenders. In *Proceedings of the SIGKDD*. 138–147.

[9] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the EMNLP*. 1724–1734.

[10] Yang Deng, Yaliang Li, Fei Sun, Bolin Ding, and Wai Lam. 2021. *Unified Conversational Recommendation Policy Learning Via Graph-Based Reinforcement Learning*. 1431–1441.

[11] 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 NAACL*. 4171–4186.

[12] Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the EMNLP*. 1295–1309.

[13] Zuohui Fu, Yikun Xian, Yaxin Zhu, Shuyuan Xu, Zelong Li, Gerard de Melo, and Yongfeng Zhang. 2021. HOOPS: Human-in-the-loop graph reasoning for conversational recommendation. In *Proceedings of the SIGIR*. 2415–2421.

[14] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2021. Advances and challenges in conversational recommender systems: A survey. *AI Open* 2 (2021), 100–126.

[15] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information tion and maximization. In *Proceedings of the ICLR*.

[16] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *Proceedings of the ICLR*.

[17] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging meta-path based context for top- n recommendation with a neural co-attention model. In *Proceedings of the SIGKDD*. 1531–1540.

[18] Chenhao Hu, Shuhua Huang, Yansen Zhang, and Yubao Liu. 2022. Learning to infer user implicit preference in conversational recommendation. In *Proceedings of the SIGIR*. 256–266.

[19] Nguyen Quoc Viet Hung, Huynh Huu Viet, Nguyen Thanh Tam, Matthias Weidlich, Hongzhi Yin, and Xiaofang Zhou. 2017. Computing crowd consensus with partial agreement. *TKDE* 30, 1 (2017), 1–14.

[20] Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph. In *Proceedings of the EMNLP*. 725–736.

[21] Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y-Lan Boureau, and Jason Weston. 2019. Recommendation as a communication game: Self-supervised bot-play for goal-oriented dialogue. In *Proceedings of the EMNLP-IJCNLP*. 1951–1961.

[22] J. F. Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *TOIS* 2, 1 (1984), 26–41.

[23] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the WSDM*. 304–312.

[24] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive path reasoning on graph for conversational recommendation. In *Proceedings of the SIGKDD*. 2073–2083.

[25] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Proceedings of the NeurIPS*.

[26] Yang Li, Tong Chen, Peng-Fei Zhang, and Hongzhi Yin. 2021. Lightweight self-attentive sequential recommendation. In *Proceedings of the CIKM*. 967–977.

[27] Lizi Liao, Yunshan Ma, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2018. Knowledge-aware multimodal dialogue systems. In *Proceedings of the ACM MM*. 801–809.

[28] Allen Lin, Ziwei Zhu, Jianling Wang, and James Caverlee. 2023. Enhancing user personalization in conversational recommenders. In *Proceedings of the WWW*.

[29] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *Proceedings of the ICLR*.

[30] Z Liu, H Wang, ZYu Niu, Hua Wu, W Che, and T Liu. 2020. Towards conversational recommendation over multi-type dialogs. In *Proceedings of the ACL*. 1036–1049.

[31] Wenchang Ma, Ryuichi Takanobu, Minghao Tu, and Minlie Huang. 2020. Bridging the gap between conversational reasoning and interactive recommendation. Retrieved from https://arXiv:2010.10333

[32] 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 ACL*. 845–854.

[33] Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the EMNLP*. 2241–2252.

[34] Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the ACL*. 6140–6150.

[35] Yunke Qu, Tong Chen, Xiangyu Zhao, Lizhen Cui, Kai Zheng, and Hongzhi Yin. 2023. Continuous input embedding size search for recommender systems. In *Proceedings of the SIGIR*. 708–717.

[36] Alec Radford, Jeffrey Wu, Rewon Child, Dvid Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *Proceedings of the OpenAI Blog*.

[37] Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Zi Huang, and Kai Zheng. 2021. Learning to ask appropriate questions in conversational recommendation. In *Proceedings of the SIGIR*. 808–817.

[38] Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Nguyen Quoc Viet Hung, Zi Huang, and Xiangliang Zhang. 2020. CRSAL: Conversational recommender systems with adversarial learning. *TOIS* 38, 4 (2020).

[39] Rajdeep Sarkar, Koustava Goswami, Mihael Arcan, and John P. McCrae. 2020. Suggest me a movie for tonight: Leveraging knowledge graphs for conversational recommendation. In *Proceedings of the COLING*. 4179–4189.

[40] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the SIGIR*. 253–260.

[41] 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 ESWC*, Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam (Eds.). 593–607.

[42] Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the EMNLP*. 4615–4629.

[43] Fan-Yun Sun, Jordon Hoffman, Vikas Verma, and Jian Tang. 2020. InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *Proceedings of the ICLR*.

[44] Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the EMNLP-IJCNLP*. 2380–2390.

[45] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *Proceedings of the SIGIR*. 235–244.

[46] Quan Tu, Shen Gao, Yanran Li, Jianwei Cui, Bin Wang, and Rui Yan. 2022. Conversational recommendation via hierarchical information modeling. In *Proceedings of the SIGIR*. 2201–2205.

[47] Huizhao Wang, Guanfeng Liu, An Liu, Zhixu Li, and Kai Zheng. 2019. DMRAN: A hierarchical fine-grained attention-based network for recommendation. In *Proceedings of the IJCAI*. 3698–3704.

[48] Huizhao Wang, Guanfeng Liu, Yan Zhao, Bolong Zheng, Pengpeng Zhao, and Kai Zheng. 2019. Dmfp: A dynamic multi-faceted fine-grained preference model for recommendation. In *Proceedings of the ICDM*. 608–617.

[49] Qinyong Wang, Hongzhi Yin, Tong Chen, Zi Huang, Hao Wang, Yanchang Zhao, and Nguyen Quoc Viet Hung. 2020. Next point-of-interest recommendation on resource-constrained mobile devices. In *Proceedings of the WWW*. 906–916.

[50] Qinyong Wang, Hongzhi Yin, Tong Chen, Junliang Yu, Alexander Zhou, and Xiangliang Zhang. 2022. Fast-adapting and privacy-preserving federated recommender system. *VLDB J.* (2022), 1–20.

[51] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the SIGKDD*. 950–958.

[52] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the SIGKDD*. 950–958.

[53] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. *AAAI* 33, 01 (2019), 5329–5336.

[54] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the SIGKDD*. 1929–1937.

[55] Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve J. Young. 2017. Latent intention dialogue models. In *Proceedings of the ICML*. 3732–3741.

[56] Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the EMNLP*. 564–573.

[57] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete KBs with knowledge-aware reader. In *Proceedings of the ACL*. 4258–4264.

[58] 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 COLING*. 5288–5308.

[59] Yi-Ting Yeh and Yun-Nung Chen. 2019. QAInfomax: Learning robust question answering system by mutual information maximization. In *Proceedings of the EMNLP-IJCNLP*. 3370–3375.

[60] Tong Yu, Yilin Shen, and Hongxia Jin. 2019. A visual dialog augmented interactive recommender system. In *Proceedings of the SIGKDD*. 157–165.

[61] Bolin Zhang, Zhiying Tu, Yangqin Jiang, Shufan He, Guoqing Chao, Dianhui Chu, and Xiaofei Xu. 2021. DGPF: A dialogue goal planning framework for cognitive service conversational bot. In *Proceedings of the ICWS*. 335–340.

[62] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the ACL*. 2204–2213.

[63] Shijie Zhang, Hongzhi Yin, Tong Chen, Zi Huang, Lizhen Cui, and Xiangliang Zhang. 2021. Graph embedding for recommendation against attribute inference attacks. In *Proceedings of the WWW*. 3002–3014.

[64] Xiaoyu Zhang, Xin Xin, Dongdong Li, Wenxuan Liu, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. 2023. Variational reasoning over incomplete knowledge graphs for conversational recommendation. In *Proceedings of the WSDM*. 231–239.

[65] 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 CIKM*. 177–186.

[66] Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Zhiping Shi, Hui Xiong, and Qing He. 2020. Relational graph neural network with hierarchical attention for knowledge graph completion. *AAAI* 34, 05 (2020), 9612–9619.

[67] Shangfei Zheng, Hongzhi Yin, Tong Chen, Quoc Viet Hung Nguyen, Wei Chen, and Lei Zhao. 2023. DREAM: Adaptive reinforcement learning based on attention mechanism for temporal knowledge graph reasoning. In *Proceedings of the SIGIR*. 1578–1588.

[68] Jinfeng Zhou, Bo Wang, Zhitong Yang, Dongming Zhao, Kun Huang, Ruifang He, and Yuexian Hou. 2022. CR-GIS: Improving conversational recommendation via goal-aware interest sequence modeling. In *Proceedings of the COLING*. 400–411.

[69] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the SIGKDD*. 1006–1014.

[70] Yu Zhu, Yu Gong, Qingwen Liu, Yingcai Ma, Wenwu Ou, Junxiong Zhu, Beidou Wang, Ziyu Guan, and Deng Cai. 2019. Query-based interactive recommendation by meta-path and adapted attention-GRU. In *Proceedings of the CIKM*. 2585–2593.

Received 26 April 2023; revised 4 November 2023; accepted 23 November 2023