KECRS: Towards Knowledge-Enriched Conversational Recommendation System

Tong Zhang, Yong Liu, Peixiang Zhong, Chen Zhang, Hao Wang, Chunyan Miao

1 School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore
2 Alibaba-NTU Singapore Joint Research Institute, NTU, Singapore
3 Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, NTU, Singapore
4 Alibaba Group, China

{tong24,stephenliu,peixiang001,ascymiao}@ntu.edu.sg, zhangchen010295@163.com, cashenry@126.com

ABSTRACT
The chit-chat-based conversational recommendation systems (CRS) provide item recommendations to users through natural language interactions. To better understand user’s intentions, external knowledge graphs (KG) have been introduced into chit-chat-based CRS. However, existing chit-chat-based CRS usually generate repetitive item recommendations, and they cannot properly infuse knowledge from KG into CRS to generate informative responses. To remedy these issues, we first reformulate the conversational recommendation task to highlight that the recommended items should be new and possibly interested by users. Then, we propose the Knowledge-Enriched Conversational Recommendation System (KECRS). Specifically, we develop the Bag-of-Entity (BOE) loss and the infusion loss to better integrate KG with CRS for generating more diverse and informative responses. BOE loss provides an additional supervision signal to guide CRS to learn from both human-written utterances and KG. Infusion loss bridges the gap between the word embeddings and entity embeddings by minimizing distances of the same words in these two embeddings. Moreover, we facilitate our study by constructing a high-quality KG, i.e., The Movie Domain Knowledge Graph (TMDKG). Experimental results on a large-scale dataset demonstrate that KECRS outperforms state-of-the-art chit-chat-based CRS, in terms of both recommendation accuracy and response generation quality.

CCS CONCEPTS
• Computing methodologies → Natural language generation;
• Information systems → Recommender systems.

KEYWORDS
Conversational Recommendation System, Dialogue System, Knowledge Graph, Deep Learning

ACM Reference Format:
Tong Zhang, Yong Liu, Peixiang Zhong, Chen Zhang, Hao Wang, Chunyan Miao. 2018. KECRS: Towards Knowledge-Enriched Conversational Recommendation System. In Woodstock ’18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3122445.3122456

1 INTRODUCTION
Recommendation system has been widely applied on e-commerce platforms to provide personalized services to customers [26]. In general, the recommendation system matches the user’s preferences with the items based on her historical behaviors, e.g., clicking history, and rating history. Although existing recommendation methods can usually achieve satisfied recommendation accuracy, they still suffer from the cold-start problem, where no historical observation is available for new users [1]. Moreover, user’s demands may also vary over time. Thus, it is very hard for traditional recommendation systems to capture user’s dynamic interests timely [10]. The recently emerging conversational recommendation system (CRS) becomes an appealing solution to solve these two problems. A CRS can interact with users by natural language and obtain users’ explicit feedback timely for better understanding users’ dynamic interests. Previous studies about CRS can be classified into two main categories: 1) attribute-based CRS, and 2) chit-chat-based CRS [10]. The attribute-based conversational recommendation methods [3, 4, 15, 16, 29, 35, 40] mainly focus on the recommendation task. They explore the user’s preferences on different item attributes to retrieve the items matching the user’s interests. Their objective is to optimize the recommendation accuracy and the number of interaction rounds between the user and the recommender agent. Differing from these attribute-based methods, the chit-chat-based conversational recommendation methods [2, 8, 17, 20, 25, 38, 39] naturally integrate the recommendation task and the response generation task. In real application scenarios, more natural and human-like dialogues may improve user’s experiences and make users more engaged [33]. Thus, we focus on the chit-chat-based CRS in this work.

Generally, a chit-chat-based CRS is composed of two main modules, i.e., the recommendation module and the response generation module. The recommendation module firstly explores the user’s intentions and recommends a list of items that meet her interests. Then, the response generation module generates appropriate responses based on the recommended item list and the dialogue context, which includes the user’s timely feedback. To support
the research about chit-chat-based CRS, the pioneer work [17] releases a large conversational recommendation dataset in the movie domain, namely Recommendations through DIALog (REDIAL). However, a conversation may only contain several utterances and lack sufficient contextual information. As an example conversation in Figure 1, if we have no prior knowledge that A Nightmare on Elm Street and The Last House on the Left are both thrillers, we cannot recommend another thriller (e.g., Happy Death Day) to the user. Besides, an ideal CRS should provide not only accurate recommendations but also informative responses that contain information about the recommended items (e.g., a new “scary movie”). Based on these concerns, Chen et al. [2] and Zhou et al. [38] exploit the knowledge graph (KG) to improve the performances of CRS.

Although these studies have made progress in CRS, they still suffer from the following limitations:

- Previous studies such as [2, 38] treat all items mentioned by the recommender as recommendations. Thus, they focus on the item mention prediction task instead of the item recommendation task. Under the settings of these studies, items may be repetitively recommended by the recommender agent, which may hurt the user’s experiences.

- Previous works are more likely to generate dull and generic responses. To generate more diverse responses, external KGs (e.g., DBpedia and ConceptNet) are usually introduced into the CRS [2, 38]. As the construction of existing conversational recommendation dataset is not based on the external KGs, there exists a big semantic gap between the information in conversation utterances and the knowledge in KGs. For example, [2, 38] use DBpedia subgraph as the external KG and REDIAL as the conversational recommendation dataset. Through our data analysis, REDIAL conversations only mention 8% of entities in DBpedia subgraph excluding movie entities. At the same time, each response by the recommender in REDIAL only contains 0.25 entities, on average. In this case, KG entities are hard to be generated in responses by previous models. The effect of the KG in the response generation module may also be limited. Moreover, KG entities’ embeddings are learned in the recommendation module while words embeddings are learned in the generation module [2, 10, 38]. The representation space gap between these two embeddings may also limit the capability of the models to generate KG entities in responses. However, existing methods mostly regard the feature obtained from the recommendation module just as an additional feature of the response generation module [2, 38]. They do not model the relationship among different spaces explicitly.

- KGs used in the previous work [2, 25, 38] are noisy and incomplete. They mostly extract a subset of an open domain KG (e.g., DBpedia and ConceptNet) and then use it in a specific domain (e.g., movie domain). These KGs may contain irrelevant information and lose high-order neighbors, which limits their contributions to both the recommendation and response generation module.

To this end, in this paper, we first reformulate the conversational recommendation task to highlight that recommended items need to be new and possibly interested in by users. Using KG as external knowledge, we propose the Knowledge-Enriched Conversational Recommendation System (KECRS), which can generate both more informative responses and more accurate recommendations. Specifically, we propose Bag-of-Entity (BOE) loss and infusion loss to generate more diverse and informative responses. BOE loss provides an additional supervision signal to guide the model to generate knowledge-enriched responses not only from utterances in training data but also from the neighboring entities of recommended items in KG. Infusion loss bridges the representation space gap between words in the generation module and entities in the recommendation modules by minimizing distances of the same entities in different spaces. With these two losses, entities related to recommended items are more likely to be generated in responses. Thus, the diversity and informativeness of responses can be highly improved.

Moreover, we facilitate our study by constructing a high-quality KG, The Movie Domain Knowledge Graph (TMDKG) related to the conversational recommendation dataset REDIAL [17]. Extensive experiments demonstrate that the proposed two losses help improve model performance and the proposed KECRS model outperforms state-of-the-art CRS models on a large-scale public dataset in terms of both recommendation accuracy and response quality. In addition, TMDKG is able to improve the performances of both the KECRS model and baseline methods.

2 RELATED WORK

Traditional recommendation systems may have cold-start problems for new users and be hard to capture users’ dynamic preferences. To tackle these problems, different methods have been proposed, e.g., introducing external knowledge [19, 34] and analyzing review data [11, 18]. However, these methods are mostly static or have limited assumptions [7]. With the developments of dialogue systems in open-domain [6, 13, 37] and task-oriented [22, 24, 31] tasks, conversational recommendation systems become an appealing solution to capture users’ dynamic preferences.

One category of conversational recommendation systems are attribute-based conversational recommendation systems [3, 4, 15, 16, 29, 35, 40]. Most studies in this category focus on how to provide high-quality recommendations within the shortest number of conversation turns but do not pay much attention to generate human-like responses. Christakopoulou et al. [4] first propose a
contextual bandit-based method to develop the conversational recommendation system and then propose an RNN-based method \[3\], which jointly optimize attribute and item prediction tasks. But these studies restrict the conversation to two turns, i.e., one turn for asking and the other for recommending. Sun and Zhang \[29\] extend it to the multi-turn conversations but still remain the single-round recommendation setting. Following this setting, Zhang et al.\[35\] propose the Multi-Memory Network architecture, which conducts question prediction and search in a parallel manner. However, in practical deployments, users may reject a recommendation sometimes, which is ignored by the previous single-round recommendation setting. Thus, Lei et al. \[15, 16\] adopt the multi-round setting, where a conversational recommendation system can conduct multi-turn conversations and do recommendations multi-times.

The other category of conversational recommendation methods are chit-chat-based conversational recommendation systems \[2, 8, 17, 20, 25, 38, 39\]. Most studies in this category focus on both giving accurate recommendations and generating natural and human-like responses. As there is no publicly available large-scale dataset consisting of real-world dialogues centered around recommendations, Li et al. \[17\] release a conversational recommendation dataset in the movie domain and propose an HRED-based \[28\] baseline model. As it is hard to understand user’s intentions only from utterances, Chen et al. \[2\] introduce KG into Chit-chat-based CRS and propose a knowledge-based recommender dialog system. Based on \[2\], Sarkar et al. \[25\] explore the performances of using different sizes of KGs in the recommendation module. To better understand the user’s preferences, Zhou et al. \[38\] leverage the entity-oriented KG (i.e., DBpedia) and the word-oriented KG (i.e., ConceptNet). To make the recommendation proactively and naturally, Liu et al. \[20\] and Zhou et al. \[39\] use topics to guide dialogue from non-recommendation to recommendation and propose the topic-guided conversational recommendation task.

### 3 PRELIMINARY

The conversational recommendation task studied in this work consists of two sub-tasks, i.e., the recommendation task and the response generation task. The objective of the recommendation task is to recommend items that are new and possibly interesting to users. However, previous works \[2, 38\] consider all items mentioned by recommenders as recommendations. This setting will cause the following two issues. Firstly, the recommendation models are more likely to generate repetitive items, which may hurt user’s experiences. As shown in Table 1, almost half of the Top-1 recommended items of KBRD \[2\] and KGSF \[38\] have been mentioned or recommended before. However, this ratio in the REDIAL dataset is only 8.5%. Thus, if we do not constrain the recommendation setting, repetitive items may keep being recommended, which may highly hurt user’s experiences. Secondly, it is unreasonable to treat the items disliked by the user as correct item recommendations. In the experiments, around 15% of items hit by KBRD and KGSF at Top-1 are not liked by users. If we treat them as correct recommendations, it conflicts with the objective of recommendation systems and makes evaluation metrics higher than practical effect. For the response generation task, it aims to generate responses given the dialogue contexts and the recommended item list.
Table 1: Statistics of Top-1 recommended items of KBRD [2] and KGSF [38] and items mentioned by the recommender in the REDIAL [17] dataset.

| Model   | Items recommended at Top-1 | Items in REDIAL |
|---------|-----------------------------|-----------------|
|         | New | Repetitive | New | Repetitive |
| KBRD    | 46.0% | 54.0% | 91.5% | 8.5% |
| KGSF    | 55.9% | 44.1% |               |          |

Following [2, 25, 38], we also introduce the KG as the external knowledge to understand the user’s preferences and generate more informative responses. However, KGs used in the previous studies [2, 25, 38] are mostly the sub-graph of open-domain KGs, e.g., DBpedia and ConceptNet, which may contain irrelevant information and lose high-order neighbors of an entity. To alleviate these issues, we build a high-quality KG, i.e., TMDKG, in the movie domain. TMDKG is constructed from the information from The Movie Database (TMDb)\(^1\), which is a community built movie and TV database. More details about TMDKG can be found in appendix A.1.

Formally, Let \(X = \{x_1, x_2, ..., x_n\}\) denotes the utterances of a conversation, where \(n\) denotes the number of conversation histories. Let \(G = \{(u_i, r, v_i) | u_i, v_i \in V, r \in R\}\) denote the KG, where each triplet \((u_i, r, v_i)\) describes that there is a relationship \(r\) between the head entity \(u_i\) and the tail entity \(v_i\). The conversational recommendation task can be formulated as learning two functions \(f(X, G)\) and \(g(X, G, f(X, G))\). Here, the function \(f(X, G)\) predicts new and user possibly interesting items based on contexts \(X\) and the KG \(G\), and the function \(g(X, G, f(X, G))\) generates human-like responses given contexts \(X\), the KG \(G\), and the list of items recommended by \(f(X, G)\). During the inference phase, both \(f\) and \(g\) are integrated to generate a natural language response correlated to the context.

4 MODEL STRUCTURE

In this section, we present the proposed Knowledge-Enriched Conversational Recommendation System (KECRS). As shown in Figure 2, KECRS consists of three main components: 1) knowledge graph encoding module, 2) recommendation module, and 3) response generation module. Next, we introduce the details of each component.

4.1 Knowledge Graph Encoding Module

As relation semantics are important to model the similarity of two movies, we adopt R-GCN [27] to embed the structural and relational information in \(G\) to learn entity representations. Specifically, at the \((l + 1)\)-th layer, the representation of an entity \(i\) in \(G\) is defined as,

\[
h_i^{(l+1)} = \sigma \left( \sum_{r \in R} \sum_{j \in N_r^i} \frac{1}{c_{i,r}} \cdot \mathbf{W}_r (h_j^{(l)} + W_0 h_i^{(l)}) \right),
\]

where \(h_i^{(l+1)} \in \mathbb{R}^{d_k}\) denotes the embedding of the entity \(i\) at \((l + 1)\)-th layer, and \(d_k\) denotes the dimension of the entity embedding at \((l+1)\)-th layer. \(N_r^i\) denotes the neighborhood set of entity \(i\) under relation \(r \in R\). \(\mathbf{W}_r^{(l)}\) is a learnable relation-specific transformation matrix for the embedding of neighboring nodes under relation \(r\), and \(W_0^{(l)}\) is a learnable transformation matrix for the self-connection relation of the embedding of entity \(i\). \(c_{i,r}\) is the normalization constant which is \(|N_r^i|\) here.

After perform \(L\) layers, we obtain multiple representations for each entity \(i\), namely \(\{h_i^{(0)}, h_i^{(1)}, ..., h_i^{(L)}\}\). To integrate different depth information, we apply the layer-aggregation mechanism proposed in [32] to compute the representation \(h_i\) as follows,

\[
h_i = \mathbf{W}_h ((h_i^{(0)}, h_i^{(1)}, ..., h_i^{(L)})) + b_h,
\]

where \([\cdot]\) denotes the concatenation operation, \(\mathbf{W}_h \in \mathbb{R}^{d_f \times (L+L_d)}\) and \(b_h \in \mathbb{R}^{d_f}\) are learnable parameters. Finally, we can obtain the hidden representations of all entities in \(G\), which is denoted by \(H \in \mathbb{R}^{(|V| \times d_f)}\).

4.2 Recommendation Module

To exploit KG for understanding the user’s intentions, we firstly map the conversation context \(X\) to an entity sequence \(E\) by simply checking each entity in the KG. After looking up the hidden representations of entities in \(E\) from \(H\), we can obtain the matrix of entities’ hidden representation \(H_E \in \mathbb{R}^{|E| \times d_f}\). Then, we apply the self-attentive mechanism [2, 38] on \(H_E\) to obtain the hidden representation \(e_E\) of \(E\) as follows,

\[
e_E = \alpha E_H,
\]

\[
\alpha = \text{softmax}(\mathbf{W}_q \tanh(\mathbf{W}_k H_E^T)),
\]

where \(\mathbf{W}_q \in \mathbb{R}^{d_q \times d_f}\) and \(\mathbf{W}_k \in \mathbb{R}^{d_q}\) are learnable parameters. \(\alpha \in \mathbb{R}^{|E|}\) is the importance vector of entities in the sequence \(E\).

To compute the probability over item candidates, we conduct inner product between the sequence representation \(e_E\) and the item representation \(H_I\) as,

\[
P_{rec} = \text{softmax}(e_E H_I^T),
\]

where \(H_I\) is the matrix selecting from \(H\) that only contains the hidden representation of items.

To optimize the recommendation module, we minimize the cross-entropy loss for selecting the true items from a list of items. Specifically, the loss \(L_{rec}\) is defined as follows,

\[
L_{rec} = -\mathbb{E}_{m \in M} \left[ \mathbb{E}_{(E, y_E) \in m} \sum_{i=1}^{N} y_E \log(P_{rec}(i)) \right],
\]

where \(M\) denotes the set of conversations, \(E\) denotes the sequence of entities, and \(y_E\) denotes the item label. \(N\) is the number of item candidates. To meet the traditional recommendation setting, we only remain the case that has new and user-liked recommendations.

4.3 Response Generation Module

The response generation module aims to generate appropriate responses to the user with the given conversation context. As Transformer [30] has shown its excellent performances in the language representation learning [5], we leverage Transformer to develop an Encoder-Decoder framework for the response generation task, considering conversation context utterances \(X\), sequence representation \(e_E\) from recommendation module, and entity representation \(H\).
Then, we sum up the word-level score at each time step and obtain the sentence-level score. After normalizing the sentence-level score with the sigmoid function, we obtain the probability for each entity in the KG occurring in responses. Thus, the sentence-level score of Bag-of-Entity can be written as follows,

$$P_{boe} = \text{sigmoid}(\sum_{j=1}^{L} P_{res}(y_j | X, y_0, y_1, ..., y_{j-1})),$$

where $L$ is the length of generated sentence, $X$ denotes the context utterance, and $y_j$ denotes the $j$-th word in ground truth response.

Here, the target entities are the one-hop neighbors of recommended items in the KG. Thus, the BOE loss is defined as follows,

$$L_{BOE} = -\mathbb{E}_{m \in M}[\mathbb{E}(X, y_{1-hop}) \in M \sum_{i=1}^{N} y_{1-hop} \log(P_{boe}(i))],$$

where $M$ denotes the set of conversations, $X$ denotes the context utterance, $N$ denotes the number of entities in the KG, and $y_{1-hop}$ denotes the label of the recommended items' one-hop neighbors in the KG.

### 4.3.4 Embedding Alignment

In practice, there usually exists a gap between the representation space of words in the generation module and the representation space of entities in the recommendation module. To align the embeddings in these two spaces, we propose the infusion loss. The core idea of infusion loss is to minimize the distance between the same word in these two hidden spaces. As a conversation may contain a large number of words, it is very time-consuming to enumerate all the word-entity pairs. Thus, we use the hidden representation $c_E$ to represent the entity sequence of the conversation and calculate the similarity between $c_E$ and the word embedding $W_{Em}$ of the vocabulary $V_{oc}$ as follows,

$$S_{similarity} = \phi'(c_E)W_{Em}^T + b_{res},$$

where $\phi'(\cdot)$ is the linear function used to align the dimension of $c_E$ and $W_{Em},$ and $b_{res} \in \mathbb{R}^{V_{oc}}$ is a learnable bias vector. Then, the infusion loss can be defined as follows,

$$L_{infuse} = -\mathbb{E}_{m \in M}[\mathbb{E}(E, d_E) \in M \|S_{similarity} - d_E\|].$$

where $M$ denotes the set of conversations, $E$ denotes the sequence of entities, and $d_E \in \{0, 1\}^{V_{oc}}$ denotes the real distribution of entities in the vocabulary. Specifically, if an entity $e$ exists in $E,$ the value of $d_E$ at the index of $e$ is 1, otherwise is 0. $\| \cdot \|$ denotes the $L_2$ distance between two vectors.

Finally, to learn the parameters of generation module, we minimize the following objective function:

$$L_{gen-all} = L_{gen} + \lambda_1 L_{BOE} + \lambda_2 L_{infuse},$$

where $\lambda_1$ and $\lambda_2$ are two hyper-parameters that can be selected by cross-validation. In testing procedure, the probability distribution over the vocabulary at time step $j$ is calculated as follows,

$$P_{all} = P_{res}(y_j | X, y'_0, y'_1, ..., y'_{j-1}) + \lambda_3 P_{boe}(y_j | x, y'_0, y'_1, ..., y'_{j-1}),$$

where $x$ denotes the context utterance, $y'_0, y'_1, ..., y'_{j-1}$ denotes the predicted sequence before time step $j,$ and $\lambda_3$ is a hyper-parameter.
Table 2: Recommendation performances of different method based on different knowledge graphs. ∗ indicates that the improvement over the best baseline method is statistically significant with p < 0.01 using student t-test.

| Model     | KG       | Recall@K (%) | Precision@K (%) | NDCG@K (%) |
|-----------|----------|--------------|-----------------|------------|
|           | K = 1    | K = 5        | K = 10          | K = 50     | K = 10 | K = 50 | K = 10 | K = 50 |
|           |          |              |                 |            |        |        |        |        |
| REDIAL    |          | 0.41         | 7.04            | 11.69      | 27.03  | 0.55   | 1.82   | 1.51   | 0.70   | 3.73  | 5.25  | 8.60  |
|           | DBpedia  | 1.40         | 8.52            | 13.54      | 30.12  | 2.20   | 2.14   | 1.78   | 0.78   | 2.08  | 7.89  | 13.61 |
|           | TMDKG    | 2.15         | 9.24            | 14.87      | 35.63  | 2.33   | 2.24   | 1.82   | 0.85   | 2.26  | 7.54  | 11.94 |
| KBRD      |          | 1.40         | 8.73            | 13.53      | 31.84  | 1.75   | 2.11   | 1.85   | 0.80   | 1.96  | 6.94  | 10.75 |
|           | DBpedia  | 2.09         | 9.31            | 14.93      | 35.38  | 2.29   | 2.31   | 1.83   | 0.81   | 1.99  | 7.61  | 12.01 |
|           | TMDKG    | 2.13         | 7.82            | 13.68      | 34.87  | 2.13   | 1.56   | 1.35   | 0.70   | 1.51  | 6.82  | 11.59 |
| COLING20  |          | 1.91         | 9.19            | 13.42      | 32.76  | 1.96   | 1.90   | 1.49   | 0.68   | 1.96  | 7.38  | 11.57 |
|           | DBpedia  | 2.13         | 7.82            | 13.68      | 34.87  | 2.13   | 1.56   | 1.35   | 0.70   | 1.51  | 6.82  | 11.59 |
|           | TMDKG    | 2.13         | 7.82            | 13.68      | 34.87  | 2.13   | 1.56   | 1.35   | 0.70   | 1.51  | 6.82  | 11.59 |
| KGSF      |          | 1.91         | 9.19            | 13.42      | 32.76  | 1.96   | 1.90   | 1.49   | 0.68   | 1.96  | 7.38  | 11.57 |
|           | DBpedia  | 2.13         | 7.82            | 13.68      | 34.87  | 2.13   | 1.56   | 1.35   | 0.70   | 1.51  | 6.82  | 11.59 |
|           | TMDKG    | 2.13         | 7.82            | 13.68      | 34.87  | 2.13   | 1.56   | 1.35   | 0.70   | 1.51  | 6.82  | 11.59 |
| KECRS     |          | 1.52         | 8.69            | 13.63      | 31.25  | 2.11   | 2.19   | 1.78   | 0.81   | 2.11  | 6.91  | 10.78 |
|           | DBpedia  | 2.25         | 9.45            | 15.66      | 36.61  | 2.77   | 2.39   | 1.99   | 0.90   | 2.77  | 7.92  | 12.38 |
|           | TMDKG    | 2.25         | 9.45            | 15.66      | 36.61  | 2.77   | 2.39   | 1.99   | 0.90   | 2.77  | 7.92  | 12.38 |

5 EXPERIMENTAL SETTINGS
In this section, we introduce the experimental datasets, evaluation metrics, baseline methods, and the model implementation details.

5.1 Experimental Datasets
The REDIAL dataset [17] is used as the conversational recommendation dataset for the experimental evaluation. This dataset is built through Amazon Mechanical Turk (AMT). Following a series of comprehensive instructions, the AMT workers generate dialogues centered around the movie recommendation. In REDIAL dataset, there are 10,006 conversations consisting of 182,150 utterances related to 51,699 movies. Following [2, 17, 38], we split the dataset into training, validation, and testing sets using the ratio of 8:1:1.

Instead, we adopt human evaluation to measure these aspects. we randomly sample 100 multi-turn conversations from the test set and invite three annotators to score responses generated by different models from the following aspects: 1) Fluency: whether responses are fluent; 2) Relevancy: whether responses are correlated with contexts; 3) Informativeness: whether responses contain rich information of recommended items. Each aspect is rated in [0, 3], and final scores are the average of all annotators. For these evaluation metrics, the higher value indicates better performances.

5.3 Baseline Models
We compare KECRS with the following baseline methods:
- REDIAL [17]: This is a basic CRS, which is based on hierarchical recurrent encoder-decoder (HRED) architecture [28].
- Transformer [30]: This is a basic transformer model that generates responses from utterance text only and does not contain a separate recommendation module.
- KBRD [2]: This is a knowledge-based CRS that employs DBpedia to understand the user’s intentions. The response generation module is based on Transformer, where the KG information serves as a bias for generation.
- COLING20 [25]: This model is based on KBRD. It studies the influences of the size of DBpedia on the performances of CRS. Here, we use the 5-hop subgraph of DBpedia which achieves the best results in the original paper. Note that this method only has a recommendation module.
- KGSF [38]: This knowledge-based CRS exploits both entity-oriented and word-oriented KGs (i.e., DBpedia and ConceptNet) to enrich the data representations. It also introduces Mutual Information Maximization (MIM) to infuse the semantic space between two knowledge graph. The dialogue generation module in KGSF is also based on Transformer, where two KG-enriched decoders are adopted.

5.4 Implementation Details
We implement KECRS by PyTorch [23]. For the recommendation module, we set the dimensionality of entity embedding $d_f$ and $d_k$ to 200. The number of R-GCN layers is set to 2. For response generation module, we set the dimensionality of word embeddings $d_t$ and the dimensionality of hidden representations $d_{c_{es}}$ of the transformer to 300. The number of transformer encoder and decoder layers is set to 2. The maximum length of transformer input $n_{j}$ is set to 256. In the generation process, the maximum length of transformer output $n_{y}$ is set to 20. For model training, we use Adam optimizer [12] and set $\beta_1$ to 0.9, $\beta_2$ to 0.99, and $\epsilon$ to $1 \times 10^{-8}$. The learning rate is set to $3 \times 10^{-3}$ for recommendation module and
Table 3: Response generation performances of different methods. * indicates the improvement over the best baseline method is statistically significant with $p < 0.01$ using student $t$-test. AEN is short for average entity number, which denotes the average number of entity in each generated response. Dist-2,3,4 is short for Distinct-2,3,4. All human evaluation scores are from 0-3.

| Model      | Automatic Evaluation | Human Evaluation |
|------------|----------------------|-------------------|
|            | Dist-2 Dist-3 Dist-4 AEN | Fluency Relevancy Informativeness |
| REDIAL     | 0.10 0.18 0.24 0.08  | 1.92 1.62 1.05  |
| Transformer| 0.15 0.31 0.46 0.15  | 2.03 1.73 1.36  |
| KBRD       | 0.26 0.30 0.45 0.15  | 2.10 1.72 1.32  |
| KGSF       | 0.33 0.49 0.61 0.17  | 2.32 2.11 1.56  |
| KECRS      | 0.48* 0.91* 1.23* 0.34* | 2.56* 2.29* 2.18* |

$1 \times 10^{-1}$ for response generation module. The batch size is set to 128 for the recommendation module and 32 for the response generation module. Moreover, we also add the L2 regularization to avoid overfitting issues and apply gradient clipping to restrict the gradients within [0, 0.1]. The hyper-parameters $\lambda_1$, $\lambda_2$, and $\lambda_3$ are empirically set to 1.5, 0.025 and 0.1 respectively.

6 EXPERIMENTS RESULTS AND ANALYSIS

In this section, we present the details about the experimental results.

6.1 Recommendation Performances

Table 2 summarizes the recommendation performances achieved by different methods. As shown in Table 2, KBRD, COLING20, KGSF, and KECRS outperform REDIAL, because they introduce external KGs to understand the user’s intentions. By using larger subgraph of DBpedia, COLING20 achieves better performances than KBRD. KGSF usually performs better than KBRD in terms of recall and NDCG, by incorporating a word-oriented KG (i.e., ConcepNet) and an entity-oriented KG (i.e., DBpedia) to enrich the entity representations. With the knowledge from DBpedia, KECRS achieves better Precision@1 than baseline methods. This indicates that the proposed KECRS model can achieve comparable recommendation performances with baseline methods, based on the knowledge information from the open domain KG. From Table 2, we can also note that the top-1 recommendation performances of KBRD, COLING20, KGSF, and KECRS can be significantly improved, by replacing DBpedia with TMDKG. This demonstrates that the high-quality information in TMDKG can help recommendation models better understand the user’s intentions and thus make more accurate item recommendations. With the knowledge from TMDKG, the proposed KECRS model significantly outperforms all baseline methods, in terms of all the evaluation metrics. For example, based on TMDKG, KECRS outperforms KBRD, COLING20, and KGSF by 18.88%, 20.96%, and 30.05%, in terms of Precision@1, respectively.

6.2 Response Generation Performances

6.2.1 Automatic Evaluation. The automatic evaluation results of different methods in the response generation task are shown in Table 3. We can note that Transformer performs better than REDIAL, which demonstrates that Transformer is powerful to understand and generate natural language. KBRD performs better than the basic Transformer model, because it adds a vocabulary bias to fuse knowledge from DBpedia into the generated responses. Among all the baseline models, KGSF generates the most diverse responses, by exploiting both DBpedia and ConceptNet. The potential reason is that KGSF employs two additional KG-based attention layers to make the generative model focus more on items and relevant entities in DBpedia and ConceptNet. Moreover, the proposed KECRS model outperforms all baseline methods with a large margin, in terms of all automatic evaluation metrics. This demonstrates that the proposed BOE loss, infusion loss, and newly constructed TMDKG can work jointly to generate more diverse and informative responses.

6.2.2 Human Evaluation. In this work, we also perform human evaluation to study the response generation performances of different methods. The human evaluation results are summarized in Table 3. We can make similar observations as in the automatic evaluation scenario, i.e., Transformer and KBRD perform better than REDIAL, KGSF performs best among all baseline models, and KECRS performs better than KGSF with a large margin. Moreover, we can note that Fluency is relevantly higher compared to Informativeness and Relevancy for all models. This indicates that responses generated by these models are fluent and can be understood by human beings. However, responses generated by baseline models are more likely to be some generic responses, which are also called “safe responses” (e.g., I haven’t seen that one). By including additional supervision signals, aligning embeddings of word and entities, and introducing the high-quality KG, our proposed KECRS model can alleviate this issue. Overall, KECRS can understand the dialogue context and generate fluent, relevant, and informative responses.

6.2.3 Ablation Study. To better understand the effectiveness of each component in KECRS, we study the performances of the following two variants of KECRS: 1) KECRS$_{w/o}$ BOE: removing the
BOE loss by setting $\lambda_1 = 0$; 2) KECS\textsubscript{w/o BOE} removing the infusion loss by setting $\lambda_2 = 0$.

Table 4 summarizes the response generation performances of KGSF, KECS\textsubscript{w/o BOE}, KECS\textsubscript{w/o Infusion}, and KECS, in terms of Distinct n-gram (n=2,3,4) and Average Entity Number (AEN). From Table 4, we can note that KECS outperforms KECS\textsubscript{w/o BOE}. This indicates that the proposed BOE loss can help the model learn to generate responses not only from ground truth but also from the knowledge information in KG. This observation also demonstrates the lack of information about recommended items in the original dataset. Moreover, KECS\textsubscript{w/o Infusion} performs poorer than KECS. This indicates that bridging the representation space gap between the word embeddings and entity embeddings also helps improve the model performances. Compared with KGSF, both KECS\textsubscript{w/o BOE} and KECS\textsubscript{w/o Infusion} can achieve better performances in terms of most metrics, which again demonstrates the effectiveness of the proposed KECS model.

To further study the effect of the infusion loss, we draw the 2D plot of word embeddings in the response module and entity embeddings in the recommendation module after PCA in Figure 3. By using the infusion loss, two embeddings tend to cluster together, which indicates the infusion loss can bridge the gap between two representation spaces.

### 6.2.4 Parameter Sensitivity Study

$\lambda_1$ and $\lambda_2$ are two important hyper-parameters used to determine the weights of different losses when training the response generation module. We conduct experiments to study the impacts of these two hyper-parameters as shown in Figure 4. We can note that, with the increase of $\lambda_1$, the performances of KECS improve first and start to decrease when $\lambda_1$ is larger than 1.5. As the primary objective of KECS is learning from the ground truth responses instead of KG, too large $\lambda_1$ may lead to negative impacts on the performances of KECS. For $\lambda_2$, when increasing it, the infusion loss may cause the word embedding over-smooth and decrease the response generation performances achieved by KECS. $\lambda_2$ is a hyper-parameter only used in the testing phase. It is used to introduce position-irrelevant entities into the generated responses. As too large of $\lambda_2$ may hurt the fluency of generated responses, we empirically set it to 0.1.

### 6.2.5 Impacts of Knowledge Graph

Similar to the experiments in Section 6.1, we replace the DBpedia used in KBRD and KGSF by TMDKG, and also replace TMDKG used in the KECS model by DBpedia. As shown in Table 5, the response generation performances of KBRD, KGSF, and KECS, in terms of all evaluation metrics, can be improved by using TMDKG instead of DBpedia. This demonstrates that TMDKG contains more sufficient and less noisy information than DBpedia for the response generation task. This observation is also consistent with the observed patterns in the recommendation module (Section 6.1). In addition, we can also note that KECS consistently achieves better response generation performances than KBRD and KGSF in terms of all metrics, based on DBpedia and TMDKG. This demonstrates the robustness of the proposed KECS model in the response generation task.

### 6.3 Case Study

Table 6 shows a case study about the conversations generated by different models. Compared with baseline models, KECS can not only capture user’s intentions to provide the corresponding recommendations, but also produce informative responses regarding the recommended item. As shown in Table 6, when the user expresses her preferences on “A Nightmare on Elm Street (1984)” and “The Last House on the Left (1972)”, KECS can understand the user may like thrillers. Thus, KECS recommends another thriller “It (2017)” to her. While providing the item recommendation, KECS also makes the conversation more informative and natural by including “if you like horror movies with murder mystery” in the response.
Table 6: Case Study. The user shows her preferences on two thrillers. KECRS captures this information and recommends another thriller with relevant information.

| Seeker: | Hi! What kind of movies do you like? |
| Recommender: | I am looking for a movie recommendation. |
| When I was younger, I really enjoyed the *A Nightmare on Elm Street* (1984). I also enjoyed watching *The Last House on the Left* (1972). |
| Transformer: | I haven’t seen that one. |
| KBRD: | I haven’t seen that one yet. |
| KGSF: | I don’t know that one. Have you seen *Boo! A Madea Halloween* (2016)? |
| KECRS (Ours): | You’ll have to check out *It* (2017) if you like horror movies with murder mystery. |
| Human: | Oh, you like scary movies? I recently watched Happy Death Day (2017). It was really good for a new “scary movie”. |

7 CONCLUSION AND FUTURE WORK

In this paper, we first reformulate the conversational recommendation task. Then, we propose a novel Knowledge-Enriched Conversational Recommendation System (KECRS). Specifically, we develop the Bag-of-Entity (BOE) loss and the infusion loss to improve the response generation performances of the proposed KECRS model. Moreover, we facilitate our study by constructing a high-quality KG, namely TMDKG. The experimental results demonstrate that the proposed BOE loss can guide the model to generate more knowledge-enriched responses by selecting entities in KG, and the infusion loss can align the vocabulary and entity embeddings into the same space. In addition, the newly built TMDKG also helps both KECRS and baseline methods better understand the user’s preferences and generate more informative responses. Overall, KECRS usually achieves superior performances on both recommendation accuracy and response quality than state-of-the-art baselines. For future work, we would like to investigate how to use keywords to conduct the conversation from chat-chat to the recommendation [20, 39]. Moreover, we are also interested in using external knowledge (e.g., KG) to modify the REDIAL dataset and make responses more relevant to the recommended items [9].

REFERENCES

[1] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems* 46 (2013), 109–132.
[2] Qibin Chen, Junyang Lin, Yichang Zhang, Mings Ding, Yukuo Cen, Hongyang Yang, and Jie Tang. 2019. Towards knowledge-based recommender dialog system. *arXiv preprint arXiv:1908.05391* (2019).
[3] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H. Chi. 2018. Q&A: A two-stage approach toward interactive recommendation. In *SIGKDD*. 139–148.
[4] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *SIGKDD*. 815–824.
[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional Transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[6] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents.
[7] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2021. Advances and challenges in conversational recommender systems: A survey. *arXiv preprint arXiv:2103.04939* (2021).
[8] Shirley Amaghour Hayati, Dongyong Kang, Qingxiao Yang, Zhi Weiyan, Shi, and Zhou Yu. 2020. INSPIRED: Toward sociable recommendation dialog systems. *arXiv preprint arXiv:2009.14506* (2020).
[9] Dietmar Jannach and Altibash Manzoor. 2020. End-to-End learning for conversational recommendation: A long way to go?. In *RecSys*. 1789–1792.
[10] Dietmar Jannach, Altibash Manzoor, Wanli Cai, and Li Chen. 2020. A survey on conversational recommender systems. *arXiv preprint arXiv:2004.00460* (2020).
[11] Zhenyan Ji, Huayu Pi, Wei Wei, Bo Xiong, Murin Woinzik, and Roberta Dama-sevicius. 2019. Recommendation based on review texts and social communities: A hybrid model. *IEEE Access* 7 (2019), 40416–40427.
[12] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[13] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaane Hajielszur. 2019. Text generation from knowledge graphs with graph transformers. *arXiv preprint arXiv:1908.05391* (2019).
[14] Jens Lehmann, Robert Iose, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, et al. 2014. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web Journal* 6 (2014).
[15] Wenzong Lei, Xiangnan He, Yingsong Miao, Qingyun Wu, Runchang Min, Yen Kan, and Tat-Seng Chua. 2020. Estimation-action-reaction: Towards deep interaction between conversational and recommender systems. In *WSDM*. 304–312.
[16] Wenzong Lei, Gangyi Zhang, Xiangnan He, Yingsong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive path reasoning on graph for conversational recommendation. In *SIGKDD*. 2073–2083.
[17] Raymond Li, Samira Ebrahimi Kahou, Hannah Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *NeurIPS*. 9725–9735.
[18] Hongtao Liu, Wenjun Wang, Hongyan Xu, Qiyaow Peng, and Pengfei Jiao. 2020. Neural unified review recommendation with cross attention. In *SIGIR*. 1789–1792.
[19] Siwei Liu, Iadu Ounis, Craig Macdonald, and Zaiqiao Meng. 2020. A heterogeneous graph neural model for cold-start recommendation. In *SIGIR*. 2029–2032.
[20] Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Towards conversational recommendation over multi-type dialogues. *arXiv preprint arXiv:2005.03954* (2020).
[21] Shuming Ma, Xu Sun, Yizhong Wang, and Jinyang Lin. 2018. Bag-of-words as target for neural machine translation. *arXiv preprint arXiv:1805.04871* (2018).
[22] Andrea Madotto, Chien-Sheng Wu, and Pascale Fang. 2018. Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *arXiv preprint arXiv:1804.08217* (2018).
[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, and et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*. 8864–8873.
[24] Xuhui Ren, Hongchi Yin, Dong Chen, Hao Wang, Nguyen Quoc Viet Hung, Zi Huang, and Xiangliang Zhang. 2020. Crsal: Conversational recommender systems with adversarial learning. *ACM Transactions on Information Systems (TOIS)* 38, 4 (2020), 1–40.
[25] Rajdeep Sarkar, Koustava Goswami, Mihael Arcan, and John Philip McCrae. 2020. Towards Keyword-Enriched Conversations: Insights Towards Improving User Experience. In *CoRR*. 4179–4189.
[26] J. Ben Schafar, Joseph A. Konstan, and John Riedl. 2001. E-commerce recommendation applications. *Data mining and knowledge discovery 5*, 1-2 (2001), 115–153.
[27] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. *arXiv preprint arXiv:1702.07519* (2017).
[28] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. A hierarchical latent variable encoder-decoder model for generating dialogues. *arXiv preprint arXiv:1605.06069* (2016).
[29] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *SIGIR*. 235–244.
[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).
[31] Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019. Global-to-local memory pointer networks for task-oriented dialogue. *arXiv preprint arXiv:1901.04713* (2019).
[32] Keyulu Xu, Chengtao Li, Yulong Tian, Tomohiro Sonobe, Ken-ichi Kawaguchi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. *arXiv preprint arXiv:1806.05358* (2018).
[33] Zhao Yan, Nan Duan, Peng Chen, Min Zhou, Jianhui Zhou, and Zhoujun Li. 2017. Building task-oriented dialogue systems for online shopping. In *AAAI*. 1562–1569.
A APPENDIX

A.1 TMDKG Construction

To better understand speaker’s intentions and avoid noisy information in the existing open-domain knowledge graph, we build a knowledge graph in the movie domain, which is called TMDKG. TMDKG contains 6 types of entities and 15 types of relations. Table 7 shows the statistics of the TMDKG.

Data Source We collect the movie information from The Movie Database (TMDb), which is a community built movie and TV database. For each movie or TV show, TMDb contains related information like plots, actors, reviews, and etc.

Information Collection The movie entities in the TMDKG are the same as the movies mentioned in REDIAL [17], a large conversational recommendation dataset in the movie domain. We use “movie name” and “released year” as keywords to search on TMDb and keep movie genres, movie cast, movie crew, movie abstract plots, movie production companies, and department of the crew to build the TMDKG. Among the 6924 movies in REDIAL, we can match 6662 (96.2%) movies in TMDb.

Data Processing We first use the Named Entity Recognition tool of spaCy2 to extract the keywords from the movie abstract plots. Then we filter out the keywords, cast, production companies that occur less than 4 times. We keep those occurring at least 10 times for the crew and all for genres. In this way, we get all the nodes in TMDKG, shown in Table 7a. For the relationship in TMDKG, movie genres, movie cast, keywords, movie production company are viewed as 4 different relations with the movie. As the crew of movies may be from different departments, we view them as different relations with movies according to the departments they belong to. The statistics of edges of TMDKG is shown in Table 7b.

Table 7: Statistics of The Movie Domain Knowledge Graph (TMDKG)

| Node Type          | Node Number |
|--------------------|-------------|
| Genre              | 19          |
| Movie              | 6924        |
| Cast               | 1861        |
| Crew               | 3523        |
| Keywords           | 2791        |
| Production         | 704         |
| All                | 15822       |

(a) Node type and number

| Edge Type                   | Edge Number |
|-----------------------------|-------------|
| Movie-Genre                 | 15710       |
| Movie-Keyword               | 37702       |
| Movie-Cast                  | 15894       |
| Movie-Crew                  | 3784        |
| Movie-Production Department | 13485       |
| Movie-Sound                 | 18489       |
| Movie-Editing               | 3425        |
| Movie-Directing             | 3367        |
| Movie-Writing               | 2915        |
| Movie-Art                   | 6117        |
| Movie-Costume & Make-up     | 4642        |
| Movie-Camera                | 5301        |
| Movie-Visual Effect         | 2044        |
| Movie-Lighting              | 445         |
| Movie-Production Company    | 10268       |
| ALL                          | 141564      |

(b) Edge type and number

---

2https://spacy.io/