

Towards Knowledge-Based Recommender Dialog System

Qibin Chen¹, Junyang Lin², Yichang Zhang², Ming Ding¹, Yukuo Cen¹, Hongxia Yang², Jie Tang¹

¹Department of Computer Science and Technology, Tsinghua University

²DAMO Academy, Alibaba Group

{cqb19, dm18, cyk18}@mails.tsinghua.edu.cn

{junyang.ljy, yichang.zyc, yang.yhx}@alibaba-inc.com

jietang@tsinghua.edu.cn

Abstract

In this paper, we propose a novel end-to-end framework called **KBRD**, which stands for **Knowledge-Based Recommender Dialog System**. It integrates the recommender system and the dialog generation system. The dialog system can enhance the performance of the recommendation system by introducing knowledge-grounded information about users' preferences, and the recommender system can improve that of the dialog generation system by providing recommendation-aware vocabulary bias. Experimental results demonstrate that our proposed model has significant advantages over the baselines in both the evaluation of dialog generation and recommendation. A series of analyses show that the two systems can bring mutual benefits to each other, and the introduced knowledge contributes to both their performances.¹

1 Introduction

Dialog in e-commerce has great commercial potential. In conventional recommender systems, personalized recommendation is highly based on the previous actions of users, including searching, clicking and purchasing. These actions can be regarded as users' feedbacks that reflect users' interest. However, due to its implicitness, such feedback can only reflect a part of users' interest, causing inaccuracy in recommendation. Another information source about user preferences is the dialog between users and services. In such dialog, users often provide more information about their preferences. They often ask for tips or recommendation in the dialog. In this process, services can guide them to speak out their interests in order to solve users' problems and meet their requirements. Compared with the implicit feedback, the

¹Code will be available at <https://github.com/THUDM/KBRD>.

USER:	Hello!
RECOMMENDER:	What kind of movies do you like?
USER:	I am looking for a movie recommendation. When I was younger I really enjoyed the <i>A Nightmare on Elm Street (1984)</i> .
BASELINE:	Have you seen <i>It (2017)</i> ?
OURS:	I like horror movies too! Have you seen <i>Halloween (1978)</i> ?
HUMAN:	Oh, you like scary movies? I recently watched <i>Happy Death Day (2017)</i> .

Table 1: **An example of the outputs of recommender dialog systems.** The model recommends items (*italic*) while maintaining the dialog with the user. Compared with the baseline, our dialog system gives a more diverse and consistent response.

feedback from the dialog is more explicit and more related to users' preferences. Therefore, a recommender dialog system possesses high commercial potential.

A recommender dialog system can be regarded as a combination of a recommender system and a dialog system. A dialog system should respond to users' utterances with informative natural language expressions, and a recommender system should provide high-quality recommendation based on the content of users' utterances. We demonstrate an example in Table 1. In brief, a recommender dialog system should perform well in both tasks.

An ideal recommender dialog system is an end-to-end framework that can effectively integrate the two systems so that they can bring mutual benefits to one another. In this setting, information from the recommender system can provide vital information to maintain multi-turn dialog, while information from the dialog system that contains implication of users' preferences can enhance the qual-

ity of recommendation. Besides, the incorporation of external knowledge can strengthen the connections between systems and enhance their performances. Therefore, driven by the motivations, we propose a novel end-to-end framework that integrates the two systems. We name it **KBRD**, standing for **K**nowledge-**B**ased **R**ecommender **D**ialog System.

Specifically, the dialog generation system provides contextual information about items to the recommender system. For instance, for a movie recommendation system, contextual information can be director, actor/actress and genre. Thus, even with no item mentioned in the dialog, the recommender system can still perform high-quality recommendation based on the contextual information. In return, the recommender system provides recommendation information to promote the dialog, such as recommendation-aware vocabulary bias. Furthermore, we incorporate external knowledge into our framework. The knowledge graph helps bridge the gap between systems and enhances both their performances.

We conduct a series of experiments that demonstrate the effects of our framework in both the evaluation of recommendation and dialog generation. Moreover, the analyses show that dialog information effectively tackles the cold-start problem in recommendation, and the recommendation-aware vocabulary bias from the recommender system improves the quality of generated dialogs. Also, the biased words can be parts of reasons to explain the system’s decisions for recommendation.

2 Preliminary

Before we introduce our proposed framework, we provide an illustration of the basic framework of the recommender dialog system to show how the recommendation system and the dialog system are organized for end-to-end training.

2.1 Recommender System

Provided with a user’s information, a recommender system is aimed at retrieving a subset of items that meet the user’s interest from all the items. In a cold-start setting, the recommender system initially has no knowledge about the user. With the progress of the dialog, the recommender system accumulates user’s information and builds a user profile. Thus, it can provide reasonable recommendation based on the user preferences re-

flected in the conversation.

To implement an effective recommender system in this task, it is available to build a recommender system based on conventional collaborative filtering algorithms (Sarwar et al., 2001) or based on neural networks (He et al., 2017). For example, Li et al. (2018) applies a user-based autoencoder (Sedhain et al., 2015) to recommend new items based on previously mentioned items in the dialog.

2.2 Dialog System

The dialog system in the basic framework is in charge of generating multi-turn dialog with a natural language generation model. The pioneering work (Li et al., 2018) on conversational recommendation task adopted Hierarchical Recurrent Encoder Decoder (HRED) (Sordoni et al., 2015b,a; Serban et al., 2016) for this part. The HRED is an encoder-decoder framework for sequence-to-sequence learning (Sutskever et al., 2014). In the framework, an encoder receives the dialog history as input and encodes it to high-level representation, while a decoder generates responses based on the encoded representation. By recursively encoding and decoding the previous information, the system makes utterances in the multi-turn dialog.

2.3 End-to-End System

In order to perform end-to-end training, we demonstrate the combination of the recommender system and conversation system. Specifically, the input of the recommender system is constructed based on the dialog history, which is a representation of mentioned items in the dialog. The output of the recommender system P_{rec} , which is a probability distribution over the item set, can be combined with the output of the dialog system $P_{\text{dialog}} \in \mathbb{R}^{|V|}$, where V refers to the vocabulary. A switching mechanism (Gulcehre et al., 2016) controls the decoder to decide whether it should generate a word from the vocabulary or an item from the recommender output at a certain timestep.

$$P(w) = p_s P_{\text{dialog}}(w) + (1 - p_s) P_{\text{rec}}(w) \quad (1)$$

$$p_s = \sigma(w_s o + b_s) \quad (2)$$

where w represents either a word from the vocabulary or an item from the item set, o is the hidden representation in the final layer of the dialog

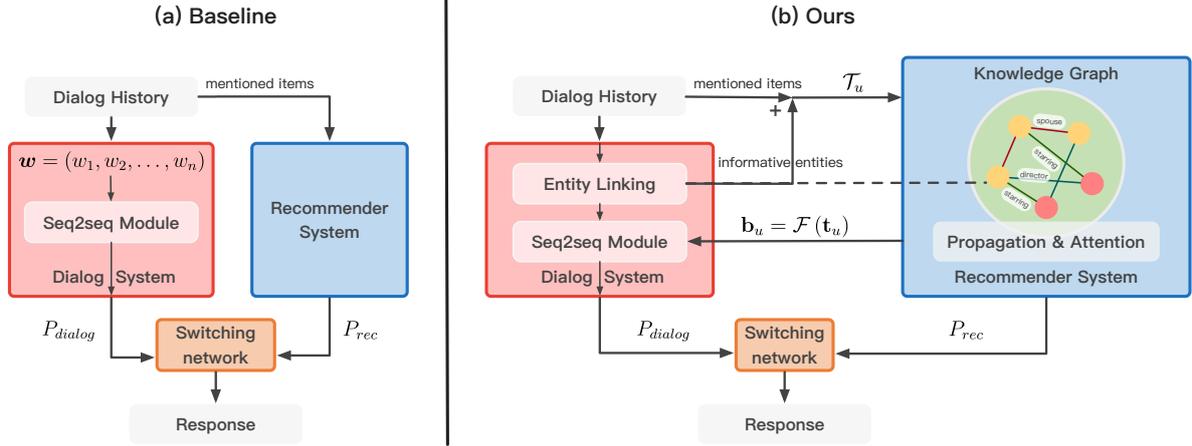


Figure 1: **Comparative illustration on modules of the existing baseline framework and our proposed KBRD framework.** (a) The connection between the recommender system and the dialog system in the baseline framework is weak. The dialog system takes the plain text of the dialog history as input and the recommender only considers mentioned items in the dialog. (b) Our framework enables interaction between the two systems. First, informative entities are linked to an external knowledge graph and sent to the recommender besides items. They are propagated on the KG via a relational graph convolutional network, enriching the representation of user interest. Second, the knowledge-enhanced user representation is sent back to the dialog system in the form of vocabulary bias, enabling it to generate responses that are consistent with the user’s interest.

system. $w_s \in \mathbb{R}^d$ and $b_s \in \mathbb{R}$ are the switcher’s parameters and σ refers to the sigmoid function. Therefore, the whole system can be trained in an end-to-end fashion.

3 Proposed Model

In this section, we introduce our proposed framework **KBRD** that integrates the recommender system and the dialog system effectively via knowledge propagation. We show how knowledge connects the two systems and how they bring mutual benefits to each other.

3.1 Dialog-Aware Recommendation with Knowledge

Recommendation of the basic framework is solely based on the mentioned items in the dialog history. Such recommendation ignores contextual information in dialog that often indicates users’ preferences.

Here we propose to make use of the dialog contents, including the non-item information, in the process of recommendation. Furthermore, to effectively recommend items from the non-item information, we introduce an external knowledge graph from DBpedia (Lehmann et al., 2015) to our system. The knowledge can build a connection between dialog contents and items.

Incorporating Dialog Contents Specifically, we have a knowledge graph \mathcal{G} consisting of triples (h, r, t) where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$. \mathcal{E} and \mathcal{R} denote the sets of entities and relations in the knowledge graph. We first match each item in the item set to entities in \mathcal{E} by name.² We then perform entity linking (Daiber et al., 2013) on dialog contents and thus informative non-item entities appearing in dialog contents are matched to \mathcal{E} .³ Therefore, we can represent a user as $\mathcal{T}_u = \{e_1, e_2, \dots, e_{|\mathcal{T}_u|}\}$, where $e_i \in \mathcal{E}$. To be more specific, it is a set of mentioned items plus non-item entities extracted from the dialog contents, linked to the knowledge graph.

Relational Graph Propagation Inspired by Schlichtkrull et al. (2018), we apply Relational Graph Convolutional Networks (R-GCNs) to encode structural and relational information in the knowledge graph to entity hidden representations. An intuition behind this is that neighboring nodes in knowledge graph may share similar features that are useful for recommendation. For exam-

²For example, a movie item “star wars” is matched to [http://dbpedia.org/resource/Star_Wars_\(film\)](http://dbpedia.org/resource/Star_Wars_(film)) in the dbpedia knowledge graph.

³Utterance “I like science fiction movies.” is associated with http://dbpedia.org/resource/Science_fiction_film. An utterance may be associated with one or multiple entities.

ple, when a user speaks of his/her preference on an actor/actress, the recommender should provide movies that have a close connection to that person. In addition, by taking different relations into consideration, the system models different types of neighbors more accurately.

Formally, at layer 0, we have a trainable embedding matrix $\mathbf{H}^{(0)} \in \mathbb{R}^{|\mathcal{E}| \times d^{(0)}}$ for nodes (i.e., entities) on the knowledge graph. Then, for each node v in \mathcal{E} at layer l , we compute:

$$h_v^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}} \sum_{w \in \mathcal{N}_v^r} \frac{1}{c_{v,r}} W_r^{(l)} h_w^{(l)} + W_0^{(l)} h_v^{(l)} \right)$$

where $h_v^{(l)} \in \mathbb{R}^{d^{(l)}}$ denotes the hidden representation of node v at the l -th layer of the graph neural network, and $d^{(l)}$ denotes the dimensionality of the representation at the layer. \mathcal{N}_v^r denotes the set of neighbor indices of node v under relation $r \in \mathcal{R}$. W_r^l is a learnable relation-specific transformation matrix for vectors from neighboring nodes with relation r . W_0^l is a learnable matrix for transforming the nodes' representation at the current layer. $c_{v,r}$ is a normalization constant that can either be learned or chosen in advance (e.g., $c_{v,r} = |\mathcal{N}_v^r|$).

For each node on the graph, it receives and aggregates the messages from its neighboring nodes after relation-specific transformation. Then it combines the information with its hidden representation to form its updated representation at the next layer.

Finally, at the last layer L , structural and relational information is encoded into the entity representation $h_v^{(L)}$ for each $v \in \mathcal{E}$. We denote the resulting knowledge-enhanced hidden representation matrix for entities in \mathcal{E} as $\mathbf{H}^{(L)} \in \mathbb{R}^{|\mathcal{E}| \times d^{(L)}}$. We omit the (L) in the following paragraphs for simplicity.

Entity Attention The next step is to recommend items to users based on knowledge-enhanced entity representations. While an item corresponds to an entity on the knowledge graph, a user may have interacted with multiple entities. Given \mathcal{T}_u , we first look up the knowledge-enhanced representation of entities in \mathcal{T}_u from \mathbf{H} , and we have:

$$\mathbf{H}_u = (h_1, h_2, \dots, h_{|\mathcal{T}_u|}) \quad (3)$$

where $h_i \in \mathbb{R}^d$ is the hidden vector of entity e_i . Here our objective is to encode this vector set of variable size to a vector of fixed size so that

we can compute the similarity between user and item. Instead of simply averaging these vectors, we choose a linear combination of the $|\mathcal{T}_u|$ vectors. Specifically, we apply self-attention mechanism (Lin et al., 2017) that takes \mathbf{H}_u as input and outputs a distribution α_u over $|\mathcal{T}_u|$ vectors:

$$\alpha_u = \text{softmax} (w_{a2} \tanh (W_{a1} \mathbf{H}_u^T)) \quad (4)$$

where $W_{a1} \in \mathbb{R}^{d_a \times d}$ is a weight matrix and w_{a2} is a vector of parameters with size d_a . The final representation of user u is computed as follows:

$$t_u = \alpha_u \mathbf{H}_u \quad (5)$$

This enables the recommender system to consider the importance of different items and non-item entities in the dialog. Finally, the output of our recommender is computed as follows:

$$P_{\text{rec}} = \text{softmax}(\text{mask}(t_u \mathbf{H}^T)) \quad (6)$$

where mask is an operation that sets the score of non-item entities to $-\infty$. The masking operation ensures that the recommendations are all items.

3.2 Recommendation-Aware Dialog

Instead of applying HRED, we introduce the Transformer framework to the dialog system in this task. Transformer (Vaswani et al., 2017) can reach significantly better performances in many tasks, such as machine translation (Vaswani et al., 2017; Ott et al., 2018), question answering (Rajpurkar et al., 2016; Yang et al., 2018; Ding et al., 2019) and natural language generation (Liu et al., 2018; Chen et al., 2019). In our preliminary experiments, we have found that Transformer can also achieve better performance than HRED in this task, and thus we apply this framework to the dialog system.

The Transformer is also an encoder-decoder framework for sequence-to-sequence learning. The Transformer encoder consists of an embedding layer and multiple encoder layers. Each encoder layer has a self-attention module and a Point-Wise Feed-Forward Network (FFN). The encoder encodes the dialog history $\mathbf{x} = (x_1, x_2, \dots, x_n)$ to high-level representations $\mathbf{s} = (s_1, s_2, \dots, s_n)$. Similarly, the Transformer decoder contains an embedding layer and multiple decoder layers with self-attention and FFN. Moreover, each of them contains a multi-head context

attention to extract information from the source-side context. The decoder generates a representation o at each decoding time step.

In order to predict a word at each decoding time step, the top layer of the decoder, namely the output layer, generates a probability distribution over the vocabulary:

$$P_{\text{dialog}} = \text{softmax}(Wo + b) \quad (7)$$

where $W \in \mathbb{R}^{|V| \times d}$ and $b \in \mathbb{R}^{|V|}$ are weight and bias parameters, and V refers to the vocabulary.

However, so far the dialog system is completely conditioned on the plain text of the dialog contents. By further introducing the recommender system’s knowledge of the items that have appeared in dialog, we guide the dialog system to generate responses that are more consistent with the user’s interests. Specifically, we add a vocabulary bias b_u to the top layer of the decoder inspired by Michel and Neubig (2018). Different from their work, b_u is computed based on the recommender system’s hidden representation of user u :

$$b_u = \mathcal{F}(t_u) \quad (8)$$

where $\mathcal{F} : \mathbb{R}^d \rightarrow \mathbb{R}^{|V|}$ represents a feed-forward neural network and t_u is the user representation in the recommendation context introduced in Equation 5.

Therefore, the computation of the top layer of the decoder becomes:

$$P_{\text{dialog}} = \text{softmax}(Wo + b + b_u) \quad (9)$$

So far, we have built an end-to-end framework that bridges the recommender system and the dialog system, which enables mutual benefits between the systems.

4 Experiments

In this section, we provide an introduction to the details of our experiments, including dataset, setting, evaluation as well as further analyses.

4.1 Dataset

REcommendations through DIALog (REDIAL) is a dataset for conversational recommendation. Li et al. (2018) collected the dialog data and built the dataset through Amazon Mechanical Turk (AMT). With enough instructions, the workers on the platform generated dialogs for recommendation on movies. Furthermore, in order to achieve and

dialog-aware recommendation, besides movies, we introduce the relevant entities, such as director and style, from DBpedia. The number of conversations is 10006 and the number of utterances is 182150. The total number of users and movies are 956 and 51699 respectively.

4.2 Setting

We implement the models in PyTorch and train on an NVIDIA 2080Ti. For the recommender, both the entity embedding size $d^{(0)}$ and the hidden representation size $d^{(l)}$ are set to 128. We choose the number of R-GCN layers $L = 1$ and the normalization constant $c_{v,r}$ to 1. For Transformer, all input embedding dimensions and hidden sizes are set to 300. During training, the batch size is set to 64. We use Adam optimizer (Kingma and Ba, 2015) with the setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$. The learning rate is 0.003 for the recommender and 0.001 for the Transformer. Gradient clipping restricts the norm of the gradients within $[0, 0.1]$.

4.3 Evaluation Metrics

The evaluation of dialog consists of automatic evaluation and human evaluation. The metrics for automatic evaluation are perplexity and distinct n-gram. Perplexity is a measurement for the fluency of natural language. Lower perplexity refers to higher fluency. Distinct n-gram is a measurement for the diversity of natural language. Specifically, we use distinct 3-gram and 4-gram at the sentence level to evaluate the diversity. As to human evaluation, we collect ten annotators with knowledge in linguistics and require them to score the candidates on the consistency with the dialog history. We sample 100 multi-turn dialogs from the test set together with the models’ corresponding responses, and require them to score the consistency of the responses.⁴ The range of score is 1 to 3.

The evaluation for recommendation is Recall@K. We evaluate that whether the top-k items selected by the recommender system contain the ground truth recommendation provided by human recommenders. Specifically, we use Recall@1, Recall@10, and Recall@50 for the evaluation.

⁴Note that we did not provide the utterances of the baseline Transformer to annotators. Based on our observation, the generations of the Transformer-based models are significantly different from those of REDIAL. In case that annotators had knowledge about models, we did not require them to score the utterances of Transformer.

Model	R@1	R@10	R@50
REDIAL	2.3±0.2	12.9±0.7	28.7±0.9
KBRD (D)	2.7±0.2	14.0±0.6	30.6±0.7
KBRD (K)	2.6±0.2	14.4±0.9	31.0±1.2
KBRD	3.0±0.2	16.3±0.3	33.8±0.7

Table 2: **Evaluation of the recommender system.** We report the results of Recall@1, Recall@10 and Recall@50 of the models ($p \ll 0.01$). **KBRD (D)** stands for only incorporating the dialog contents. **KBRD (K)** stands for only incorporating knowledge. The results demonstrate that both the interaction with the dialog system and the external knowledge are helpful for the improvement of model performance, and our proposed model reaches the best performance on the three metrics.

4.4 Baselines

The baseline models for the experiments are illustrated in the following:

- **REDIAL** This is a basic model for conversational recommendation. It basically consists of a dialog generation system based on HRED (Sordoni et al., 2015a; Serban et al., 2016), a recommendation system based on autoencoder and a sentiment analysis module.
- **Transformer** We name our implemented baseline model Transformer. It is similar to REDIAL, but its dialog generation system is based on the model Transformer (Vaswani et al., 2017). Except for that, the others remain the same.

4.5 Results

In the following, we present the results of our experiments, including the model performances in recommendation and dialog generation.

Recommendation To evaluate the effects of our recommendation system, we conduct an evaluation of Recall@K. We present the results in Table 2. From the results, it can be found that our proposed model reaches the best performances in the evaluation of Recall@1, Recall@10 and Recall@50. Furthermore, we also demonstrate an ablation study to observe the contribution of the dialog system and the introduced knowledge. It can be found that either dialog or knowledge can bring improvement to the performance of the

Model	PPL	Dist-3	Dist-4	CSTC
REDIAL	28.1	0.11	0.13	1.73
Transformer	18.0	0.27	0.39	-
KBRD	17.9	0.30	0.45	1.99

Table 3: **Automatic and human evaluation of dialog generation.** For automatic evaluation, we evaluate the perplexity (PPL) and distinct n-gram (Dist-3 and Dist-4 refer to distinct 3-gram and 4-gram respectively) of the generated dialogs. For human evaluation, we ask human annotators to evaluate the consistency (CSTC) of the generated utterances with the dialog history. Our proposed method performs the best in all evaluations compared with the baselines.

recommendation system. Their combination improves the performance the most by +0.7 Recall@1, +3.4 Recall@10 and +5.1 Recall@50, which are advantages of 30.4%, 26.4% and 17.8% respectively. This shows that the information from both sources is contributive. The dialog contains users’ preferred items as well as attributes, such as movie director and movie style, so that the system can find recommendation based on these inputs. The knowledge contains important features of the movie items so that the system can find items with similar features. Further, the combination brings an advantage even greater than sum of the two parts, which proves the effectiveness of our model.

Dialog Table 3 shows the results of the evaluation of the baseline models and our proposed method in dialog generation. In the evaluation of perplexity, Transformer has much lower perplexity (18.0) compared to REDIAL (28.1), and **KBRD** can reach the best performance in perplexity. This demonstrates the power of Transformer in modeling natural language. In the evaluation of diversity, we find that the models based on Transformer significantly outperform REDIAL from the results of distinct 3-gram and 4-gram. Besides, it can be found that **KBRD** has a clear advantage in diversity over the baseline Transformer. This shows that our model can generate more diverse contents without decreasing fluency.

As to the human evaluation, we ask human annotators to score the utterances’ consistency with their dialog history. Compared with REDIAL **KBRD** reaches better performance by +0.22 consistency score, which is an advantage of 15%. Moreover, considering the range is between 1 and 3, this is a large gap between the performances

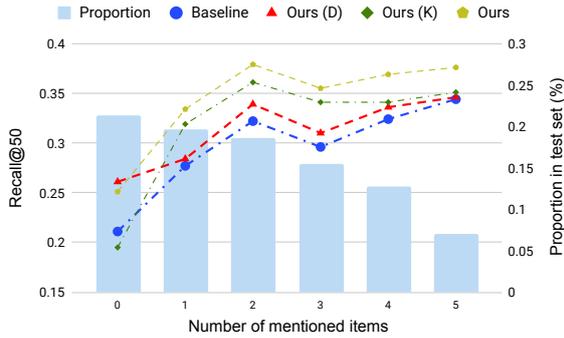


Figure 2: **Performance of the recommender system with different numbers of mentioned items.** The x-axis refers to the number of mentioned items in the dialog, the y-axis for the line chart (on the left) refers to the model performance on the Recall@50 evaluation, and the y-axis for the histogram (on the right) refers to proportion in the test set. This shows recommendation is much more difficult with few items mentioned (i.e., at the first few rounds in dialog). Leveraging dialog contents makes a great difference in this situation.

of the two models in this evaluation. To make a consistent response in a dialog, the model should understand the dialog history and better learn the user’s preference. The baseline REDIAL does not have a strong connection between the dialog system and user representation. Instead, in our framework, the recommender system provides the recommendation-aware vocabulary bias b_u , which is based on the user representation t_u , to the dialog system. Thus the dialog system gains knowledge about the user’s preference and generates a consistent response.

5 Discussion

In this section, we conduct a series of analyses to observe the effects of our proposed model. We discuss how dialog can improve the recommendation performance and how recommendation can enhance the dialog quality.

5.1 Does dialog help recommendation?

We first evaluate whether the dialog contents can benefit the recommendation system. The results of the evaluation are demonstrated in Figure 2. From the histogram in the figure, we observe that most of the dialogs contain only a few mentioned movies. The dialogs with only 0-2 mentioned movies take up a proportion of 62.8% of the whole testing dataset. Therefore, it is important for the system to perform high-quality recommendation with only a small number of mentioned movies.

This also corresponds to the classical problem “cold start” (Schein et al., 2002) in the recommender system. In real applications, we also expect that the system can perform high-quality recommendation with fewer rounds. This represents the efficiency of the recommender system, which can save users’ time and efforts.

Specifically, we demonstrate the performances of four systems in Figure 2. They are the basic framework, the one only with the interaction with the dialog system, the one only with the external knowledge and **KBRD** with both dialog and knowledge incorporation. From the figure, it can be found that while there is no mentioned item in the dialog, the baseline and the one only with knowledge perform the worst. In contrast, the two models with dialog incorporation perform significantly better. This shows that the context in the dialog contains much useful non-item information about users’ preferences, such as director, actor/actress in movie recommendation. Therefore, while there is no mentioned item, the recommender system can still perform high-quality recommendation based on the contextual information. With the increase of mentioned items, the contribution of knowledge becomes more significant than the dialog. On average, the system with both information sources performs the best. Dialog introduces contextual information and knowledge introduces movie features and structural connection with other movies.

5.2 Does recommendation help dialog?

In Section 4.5, we present the performances of the baselines and our model **KBRD** in dialog generation. It can be found that the interaction with the recommendation system can enhance the performance of the dialog system in both automatic evaluation and human evaluation. Also, an example of the responses of different models is shown in Table 1. With the dialog history, the baseline REDIAL simply uses a generic response with a recommended movie. Instead, **KBRD** has more concern about the mentioned items apart from the plain text of dialog history. The user representation from our recommender system contains such information, which is sent to the dialog system to form a vocabulary bias. With such information, **KBRD** has a better understanding of both the dialog history as well as the user’s preference, and thus generates a consistent response.

Movie	1	2	3	4	5	6	7	8
Star Wars	space	alien	sci-fi	star	sci	robot	smith	harry
The Shining	creepy	stephen	gory	horror	scary	psychological	haunted	thriller
The Avengers (2012)	marvel	superhero	super	dc	wait	batman	thor	take
Beauty and the Beast	cute	disney	animated	live	music	child	robin	kids

Table 4: **Examples of top 8 vocabulary bias.** Given a mentioned movie, we visualize words with high probability based on the recommendation-aware vocabulary bias. The examples show that the biased words have strong connections with their corresponding movies. While the vocabulary bias benefits the dialog system to generate responses that are more consistent with user interest, they also contribute to the explainability of recommendation.

To further study the effects of the recommender system on dialog generation, we display the top biased words from the vocabulary bias. Note that in **KBRD** a connection between the recommender system and dialog system is the recommendation-aware vocabulary bias b_u . To be specific, we compute the recommendation-aware bias b_u in dialog and select the components with the top-8 largest values.⁵ Then we record the corresponding words and observe whether these words are related to the mentioned movies. We present several examples in Table 4. From the table, we observe that the words are highly related to the mentioned movies. For example, when “The Shining” is mentioned, some of the top biased words are “creepy”, “gory” and “scary”, which are consistent with the style of the horror movie, and “stephen”, who is the original creator of the movie. Therefore, it can be suggested that the recommendation system conveys important information to the dialog system in the form of a vocabulary bias. Furthermore, these biased words can also serve as explicit explanation to recommendation results. From this perspective, this shows the interpretability of our model.

6 Related Work

Recommender systems aim to find a small set of items that meet users’ interest based on users’ historical interactions. Traditional recommender systems rely on collaborative filtering (Resnick et al., 1994; Sarwar et al., 2001), and recent advances in this field rely much on neural networks (Wang et al., 2015; He et al., 2017; Ying et al., 2018). To deal with the cold-start problem and the sparsity of user-item interactions which these methods usually suffer, researchers have proposed methods to incorporate external information, such as heterogeneous information networks (Yu et al., 2014), knowledge bases (Zhang et al., 2016; Wang et al., 2018a) and social networks (Jamali and

⁵After stop words filtering.

Ester, 2010). Besides accuracy, explainability is also an important aspect when evaluating recommender systems (Zhang et al., 2014; Zhang and Chen, 2018; Wang et al., 2018b).

End-to-end dialog systems based on neural networks have shown promising performance in open-ended settings (Vinyals and Le, 2015; Sordani et al., 2015b; Dodge et al., 2016; Wen et al., 2015) and goal-oriented applications (Bordes et al., 2017). Recent literature also explores the intersection of end-to-end dialog systems with other intelligence systems and creates new tasks such as visual dialog (Das et al., 2017; De Vries et al., 2017), conversational recommendation (Li et al., 2018). In particular, Li et al. (2018) collects a dataset of conversations focused on providing movie recommendations and proposes a baseline model for end-to-end training of recommender and dialog systems. Earlier studies in this field focus on different tasks such as minimizing the number of user queries (Christakopoulou et al., 2016), training the dialog agent to ask for facet values for recommendation (Sun and Zhang, 2018). Related literature can also be found in Thompson et al. (2004), Mahmood and Ricci (2009), Chen and Pu (2012), Widyantoro and Baizal (2014) and Liao et al. (2019).

7 Conclusion

In this paper, we propose a novel end-to-end framework, **KBRD**, which bridges the gap between the recommender system and the dialog system via knowledge propagation. Through a series of experiments, we show that **KBRD** can reach better performances in both recommendation and dialog generation in comparison with the baselines. We also discuss how the two systems benefit each other. Dialog information is effective for the recommender system especially in the setting of cold start, and the introduction of knowledge can strengthen the recommendation performance

significantly. Information from the recommender system that contains the user preference and the relevant knowledge can enhance the consistency and diversity of the generated dialogs.

Acknowledgements

The work is supported by NSFC for Distinguished Young Scholar (61825602), NSFC (61836013), and a research fund supported by Alibaba. The authors would like to thank Lei Li and Chang Zhou for their insightful feedback, and responsible reviewers of EMNLP-IJCNLP 2019 for their valuable suggestions. Jie Tang is the corresponding author.

References

- Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.
- Li Chen and Pearl Pu. 2012. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction*, 22(1-2):125–150.
- Qibin Chen, Junyang Lin, Yichang Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Towards knowledge-based personalized product description generation in e-commerce. *arXiv preprint arXiv:1903.12457*.
- Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 815–824. ACM.
- Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 121–124. ACM.
- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–335.
- Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5503–5512.
- Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *ACL (1)*, pages 2694–2703. Association for Computational Linguistics.
- Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander H. Miller, Arthur Szlam, and Jason Weston. 2016. Evaluating prerequisite qualities for learning end-to-end dialog systems. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 140–149.
- Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee.
- Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, pages 135–142. ACM.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.
- Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.
- Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pages 9748–9758.
- Lizi Liao, Ryuichi Takanobu, Yunshan Ma, Xun Yang, Minlie Huang, and Tat-Seng Chua. 2019. Deep conversational recommender in travel. *CoRR*, abs/1907.00710.
- 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 *5th International Conference on Learning Representations, ICLR 2017*.

- Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018*.
- Tariq Mahmood and Francesco Ricci. 2009. Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 73–82. ACM.
- Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pages 312–318.
- Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018*, pages 1–9.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392. The Association for Computational Linguistics.
- Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. **GroupLens: An open architecture for collaborative filtering of netnews**. In *CSCW '94, Proceedings of the Conference on Computer Supported Cooperative Work, Chapel Hill, NC, USA, October 22-26, 1994*, pages 175–186. ACM.
- Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW'01*, pages 285–295.
- Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM.
- 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 *European Semantic Web Conference*, pages 593–607. Springer.
- Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015*, pages 111–112.
- Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3776–3784.
- Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015a. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015*, pages 553–562.
- Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015b. A neural network approach to context-sensitive generation of conversational responses. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205.
- Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 235–244. ACM.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112.
- Cynthia A Thompson, Mehmet H Goker, and Pat Langley. 2004. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, 21:393–428.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 6000–6010.
- Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.
- Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1235–1244.
- Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018a. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 417–426. ACM.

- Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. 2018b. A reinforcement learning framework for explainable recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 587–596. IEEE.
- Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.
- Dwi H Widiantoro and ZKA Baizal. 2014. A framework of conversational recommender system based on user functional requirements. In *2014 2nd international conference on information and communication technology (ICoICT)*, pages 160–165. IEEE.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pages 2369–2380. Association for Computational Linguistics.
- Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983. ACM.
- Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norrick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 283–292. ACM.
- Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362. ACM.
- Yongfeng Zhang and Xu Chen. 2018. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192*.
- Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. ACM.