CP-Rec: Contextual Prompting for Conversational Recommender Systems

Keyu Chen, Shiliang Sun*

School of Computer Science and Technology, East China Normal University, Shanghai, China
51205901068@stu.ecnu.edu.cn, slsun@cs.ecnu.edu.cn

Abstract

The conversational recommender system (CRS) aims to provide high-quality recommendations through interactive dialogues. However, previous CRS models have no effective mechanisms for task planning and topic elaboration, and thus they hardly maintain coherence in multi-task recommendation dialogues. Inspired by recent advances in prompt-based learning, we propose a novel contextual prompting framework for dialogue management, which optimizes prompts based on context, topics, and user profiles. Specifically, we develop a topic controller to sequentially plan the subtasks, and a prompt search module to construct context-aware prompts. We further adopt external knowledge to enrich user profiles and make knowledge-aware recommendations. Incorporating these techniques, we propose a conversational recommender system with contextual prompting, namely CP-Rec. Experimental results demonstrate that it achieves state-of-the-art recommendation accuracy and generates more coherent and informative conversations.

Introduction

With the widespread applications of conversational assistants, such as Google Assistant, Apple Siri, Amazon Alexa, and Microsoft Cortana, the conversational recommender system (CRS) has received immense interest in recent years (Li et al. 2018; Chen et al. 2019; Zhou et al. 2020a,b). CRS integrates recommendation techniques into the conversational system, which can help users find desired information through interactive conversations.

In real-life scenarios, the recommendation dialogue is open-ended, where the users and the CRS interact around recommendations using free-form natural language. The users will start the conversation casually, and direct the dialogue topics to discover what they need. The dialogue may involve many phases, such as greeting, explaining, and recommending. Each phase has a certain subtask. As illustrated in Figure 1, a recommendation dialogue could be multi-task, where the system starts with chit-chat, and then seeks the user preference and makes final recommendations. Therefore, the CRS has to respond accordingly to meet users' needs and lead dialogues towards the recommendation goal.

*Corresponding author.

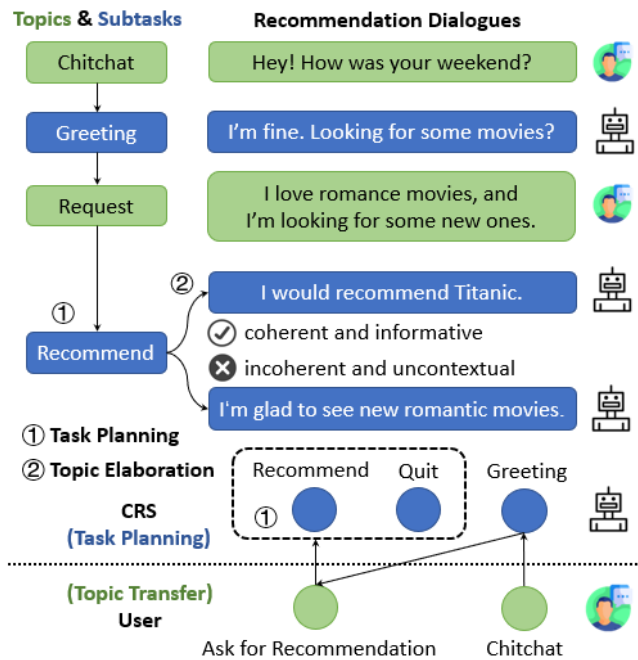

Figure 1: An illustrative example. In the recommendation dialogue, the CRS identifies the topic (①) and responds accordingly to complete each subtask (②). Task planning maintains the conversation with well-connected topics, and topic elaboration ensures that each topic can be expressed by contextual and informative sentences. Reasonable planning and proper elaboration can improve conversation coherence.

Although previous CRS methods have achieved promising results, they can hardly match the user preference while maintaining the conversation coherency. A recent study (Jannach and Manzoor 2020) on two CRS baselines, i.e., ReDial (Li et al. 2018) and KBRD (Chen et al. 2019), shows that about one-third of utterances generated by these two models are considered meaningless in the given context, and more than one-third of recommendations do not suit the assumed user preferences. It indicates that many CRS models can hardly maintain coherent conversations and accurate recommendations, especially in long, multi-task dialogues. This can be explained to some extent by their lack of mechanisms to manage topics and improve dialogue coherence and informativeness. Moreover, human dialogue research (Hirano, Higashinaka, and Matsuo 2016) shows that dialogue management, e.g., task planning and topic elaboration, is universal in human language interactions. As shown in Figure 1, the conversation with properly-planned and clearly-expressed topics is more organized and coherent. Therefore, we hypothesize that the CRS models can converse coherently if they can learn to manage chatting topics.

Two challenges arise in order to achieve effective dialogue management. The first one is how to conduct reasonable task planning in multi-task dialogues. The model should capture the user's interests to make final recommendations, and maintain the conversation with well-connected topics. The second one is how to generate in-depth replies within topics for subtask completion. The system responses should be consistent with both topics and context. Moreover, topic elaboration can promote the dialogue coherence, but it is rarely considered in existing CRS models.

Effective dialogue management means that CRS should organize conversations in a user-oriented way. We also note that the system replies in multi-task recommendation dialogue are mostly affected by the dialogue history, current topics, and user interests. Therefore, we propose CP-Rec, a conversational recommender system with contextual prompting, to tackle the above issues. First, we introduce the knowledge graph to enrich user profiles, which helps to achieve user-driven topic planning and a more accurate knowledge-aware recommendation. Second, recent advances in prompt-based learning give us new inspiration for efficient topic elaboration. Given topical, contextual, and knowledge-based prompts, CRS can reply with in-depth, coherent and informative sentences. To this end, we design a novel contextual prompting framework for joint task planning and topic elaboration. Specifically, it comprises (1) a topic controller which sequentially plans subtasks with dialogue history and user profiles, (2) a prompt search module to construct context-aware prompts, and (3) a dialogue generator. Different from traditional prompt-based dialogue systems (Madotto et al. 2021; Kasahara et al. 2022; Wang et al. 2022), we fully utilize contextual semantics and external knowledge to create continuous prompts, which enhance the system's ability to capture user preferences and generate coherent, informative dialogues. Overall, our main contributions are threefold as follows:

- We present a novel contextual prompting framework for more effective dialogue management. It incorporates dialogue history, topics, and user profiles to optimize continuous prompting representations and achieves joint task planning and topic elaboration.
- Aiming to build CRS for multi-task recommendation dialogues, we propose CP-Rec, which explicitly plans subtasks and illustrates topics via prompt learning, and better maintains the dialogue coherence and informativeness.
- Experiments on five datasets demonstrate the superior performance of our method in both recommendation and conversation tasks.

Related Work

Knowledge-Aware Recommendations In the field of e-commerce, recommender systems provide users with personalized recommendations for products or services. Traditional recommender systems are implemented by collaborative filter (Sarwar et al. 2001) or factorization machine (Rendle 2010). Recently, introducing a knowledge graph (KG) into the CRS, called knowledge-based CRS, has attracted much research attention. Compared with traditional methods, knowledge-aware recommendations utilize side information and connectivity patterns in KGs, and have better performance and explainability. Some knowledge-based CRS models improve recommendations by learning entity embeddings to enrich item representations (Chen et al. 2019; Sarkar et al. 2020; Zhou et al. 2020a; Liang et al. 2021), and other works apply multi-hop graph reasoning to provide explainable recommendations (Fu et al. 2020; Lei et al. 2020; Ma, Takanobu, and Huang 2021; Xu et al. 2020; Moon et al. 2019). Inspired by these works, we leverage the KG to enrich the user profile on the basis of dialogue history. We form a more powerful user representation to improve recommendations and enhance promptings.

Multi-Task Recommendation Dialogues CRS can be regarded as a variation of task-oriented dialogue systems, which supports users to achieve recommendation-related goals. However, it is challenging for CRS to converse fluently while completing the recommendation tasks, since dialogues are of multi-task with subtasks like greeting, requesting and recommending. To this end, many efforts have been devoted to make the CRS applicable to multi-task scenarios. TCR (Liao et al. 2022) employs a global topic control module to switch between subtasks. TG-ReDial (Zhou et al. 2020b) adapts to the topic transfer via topic threads. Some other methods adopt reinforcement learning (RL) to select high-level dialogue actions (Ren et al. 2020; Cai and Chen 2020; Chen and Sun 2021). Different from existing works, we aim to improve the model's semantic coherence and informativeness. We propose a novel dialogue management method, called contextual prompting, where reasonable task planning and clear topic elaboration work together to generate human-like responses.

Prompt-Based Learning for Language Models The GPT-3 model (Brown et al. 2020) has illustrated the few-shot capabilities of pretrained language models (PLMs). Given only a few task-oriented demonstrations as prompts, PLMs achieve comparable results in many language understanding tasks. These findings have elicited much research on prompt-based learning. Prompts can be manually designed as discrete tokens (Gao, Fisch, and Chen 2021; Jiang et al. 2020; Shin et al. 2020), or directly optimized as learnable vectors (Lester, Al-Rfou, and Constant 2021; Li and Liang 2021; Gu et al. 2022). Recent studies on prompt-based dialogue generation consider a mask language modeling (MLM) problem, where the model directly generates textual responses with given prompts. Madotto et al. (2021) adopt few-shot dialogue generation with discrete prompts. Kasahara et al. (2022) design a persona-based dialogue system via prompt-tuning. Wang et al. (2022) propose knowledge-

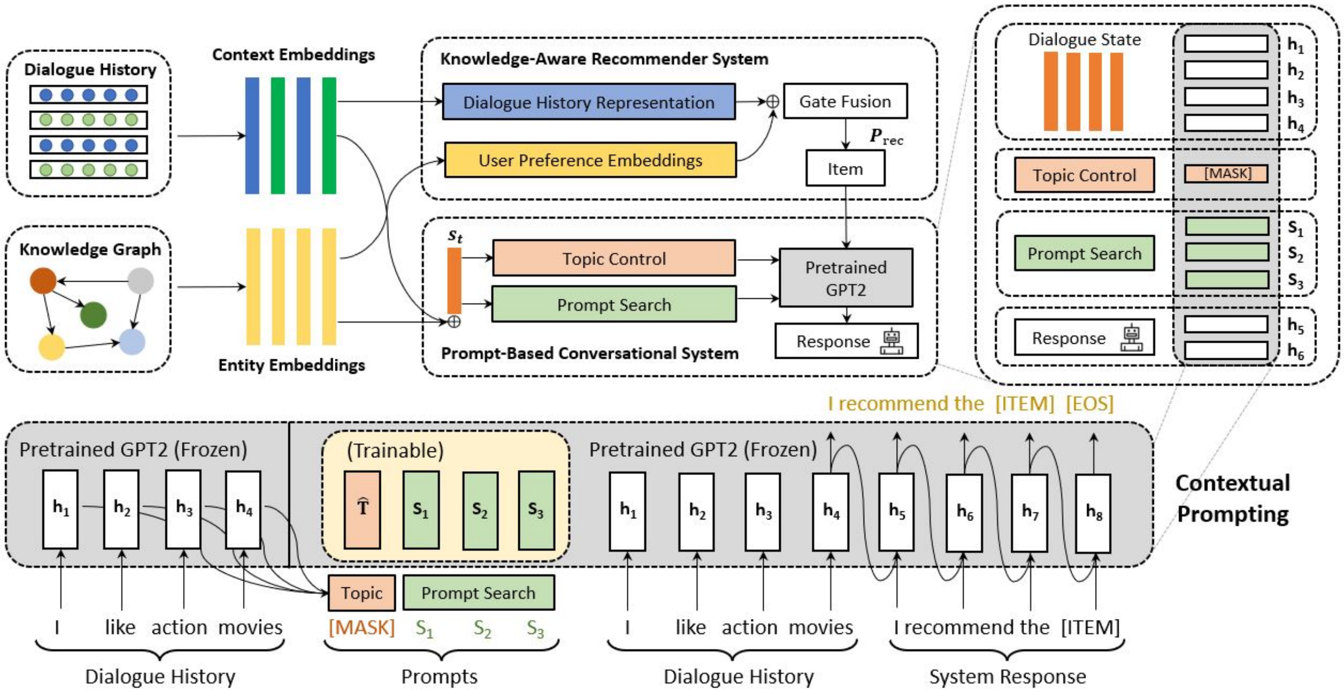

Figure 2: The overview of the proposed CP-Rec model. The knowledge-aware recommender system retrieves items according to the user preference and the dialogue history. The prompt-based conversational system consists of the contextual prompting framework with topic control and automatic prompt search.

enhanced prompting which unifies the recommendation and conversation tasks. In contrast, we leverage a context-aware prompting framework, where dialogue history, topics, and user profiles are integrated into continuous prompt encodings. Our CP-Rec tracks the user preference effectively, plans topics sequentially, and generates replies coherently.

The Proposed Model

In this section, we introduce our proposed model CP-Rec. The overview of the model is presented in Figure 2.

Our model consists of a knowledge-aware recommender system and a prompt-based conversational system. The recommender system uses context embeddings and knowledge representation of items to model user profiles. We firstly encode the dialogue history and the knowledge graph, and then compute knowledge-aware user profiles, and finally retrieve the items that match the user's preference as recommendations. The conversational system learns to control topics and optimize prompting vectors via contextual prompting. In prompt learning paradigm, the PLM used for dialogue generation is frozen. Prompts are optimized as parameters during the training process, which will be directly used to steer the frozen PLM to generate expected sentences. In the following sections, we explain the problem settings and introduce each component in detail.

Problem Settings

We define the dialogue history in the t-th dialogue turn as $\mathcal{D}_t = \{U_1, S_1, ..., U_t\}$, where U_i and S_i denote utterances of the user and the system. The topic sequence G is defined as the set of topics g_i in each U_i, namely $G = \{g_i\}_{i=1}^{t}$. The goal of the system is to (1) capture the user preference and recommend an item if necessary, and (2) predict the current topic and respond to the user. These two goals are called recommendation and conversation, which will be evaluated respectively in experiments.

Knowledge-Aware Recommender System

Context Encoder In the t-th dialogue turn, we use a pretrained BERT (Devlin et al. 2019) as the context encoder to encode the user utterance $U_t = ([\texttt{CLS}], w_1, ..., w_n)$, where w_i denotes the i-th token in U_t. According to the properties of BERT, we take the embedding of $[\texttt{CLS}]$ token as the sentence embedding, which is denoted as $\mathrm{BERT}(U_t)$. The dialogue history representation $\boldsymbol{u}_t \in \mathbb{R}^{d_u}$ is obtained by applying an LSTM over the representations of each U_t as:

$$\boldsymbol{u}_t = \mathrm{LSTM}\left(\boldsymbol{u}_{t-1}, \mathrm{BERT}(U_t)\right). \quad (1)$$

Knowledge Encoder In our model, we introduce DBpedia (Lehmann et al. 2015) as the external KG. We collect all entities in the dialogue corpus and their one-hop neighbors in DBpedia to build a knowledge subgraph $\mathcal{G}$ for training. A triple in $\mathcal{G}$ is denoted as (h, r, t), where $h, t \in \mathcal{E}$ are items from the entity set $\mathcal{E}$ and $r \in \mathcal{R}$ is an entity relation from the relation set $\mathcal{R}$. We leverage R-GCN (Schlichtkrull et al. 2018) as the knowledge encoder to learn entity representations in the extracted subgraph. The embedding of node h in the $(l+1)$-th layer is calculated as:

$$\boldsymbol{e}_h^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{\boldsymbol{e}_t\in\mathcal{N}_h}\frac{1}{Z_{h,r}}\mathbf{W}_r^l\boldsymbol{e}_t^l + \mathbf{W}^l\boldsymbol{e}_h^l\right), \tag{2}$$

where $\boldsymbol{e}_h^l \in \mathbb{R}^{d_e}$ denotes the embedding of h at the l-th layer, $\mathcal{N}_h$ is the set of neighboring nodes of h, $\mathbf{W}_r^l \in \mathbb{R}^{d_e\times d_e}$ and $\mathbf{W}^l \in \mathbb{R}^{d_e\times d_e}$ are learnable transformation matrices, $Z_{h,r}$ is the normalization factor and $\sigma(\cdot)$ is the sigmoid function. We define $\boldsymbol{R} = (\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_N)^{\mathrm{T}} \in \mathbb{R}^{N\times d_e}$ as the embedding matrix consisting of the knowledge representations of all the N items in $\mathcal{G}$.

Knowledge-Aware Recommendation We define the set of liked items mentioned by the user in the conversation as the interaction sequence, namely $I_k = \{c_i\}_{i=1}^k$. c_i denotes the i-th item the user likes, which is annotated in datasets and is aligned with an entity in $\mathcal{G}$. Assuming that each c_i contributes to the user preference to varying degrees, we calculate the preference embedding $\mathcal{P}_u$ via self-attention mechanism:

$$\mathcal{P}_u = \sum_{i=1}^k \alpha_i \cdot \boldsymbol{e}_i, \tag{3}$$

$$\alpha_i = \mathrm{softmax}(\mathbf{b}_\alpha \cdot \tanh(\mathbf{W}_R \cdot \mathbf{R}^{\mathrm{T}})). \tag{4}$$

The above $\mathbf{W}_R \in \mathbb{R}^{k\times d_e}$ and $\mathbf{b}_\alpha \in \mathbb{R}^k$ are learnable parameters, and $\boldsymbol{e}_i \in \mathbb{R}^{d_e}$ denotes the knowledge representation of c_i. Then we fuse the dialogue history embeddings and the preference embeddings to get the user profile representation $\boldsymbol{e}_u$ using gate fusion:

$$\beta = \sigma\left(\mathbf{W}_\beta \cdot (\boldsymbol{u}_t \oplus \mathcal{P}_u)\right), \tag{5}$$

$$\boldsymbol{e}_u = \beta \cdot \boldsymbol{u}_t + (1-\beta)\cdot \mathcal{P}_u, \tag{6}$$

where $\oplus$ represents the concatenation operation and $\mathbf{W}_\beta \in \mathbb{R}^{d_u+d_e}$ is a projection vector. Then the matching score $\hat{\boldsymbol{p}} \in \mathbb{R}^N$ of each item is calculated as:

$$\hat{\boldsymbol{p}} = \mathrm{softmax}(\mathbf{R}\cdot \boldsymbol{e}_u). \tag{7}$$

In practical use, multiple recommendations are allowed in our model, while in the training process, we only consider a single one. Therefore, the item with the highest matching score is selected and will be further used for dialogue generation. Suppose $\boldsymbol{p} \in \mathbb{R}^N$ denotes the ground truth vector of the recommended item, we take the cross-entropy loss as the objective function of the recommender system:

$$\mathcal{L}_{\text{rec}} = -\frac{1}{N}\sum_{i=1}^N \left[\boldsymbol{p}_i \log \hat{\boldsymbol{p}}_i + (1-\boldsymbol{p}_i)\log(1-\hat{\boldsymbol{p}}_i)\right]. \tag{8}$$

Prompt-Based Conversational System

Here we introduce our contextual prompting framework in the prompt-based conversational system. We design a topic controller to conduct reasonable task planning, which predicts the current topics in the given dialogue states. Dialogue states with user profiles encourage the system to plan topics in a user-oriented way. We further introduce prompt search to optimize prompt embeddings, where a Transformer encoder (Vaswani et al. 2017) is utilized to integrate context, topics, and user preferences into promptings. Taking topics and prompting vectors as prompts, a pretrained GPT2 (Radford et al. 2019) model generates conversations while being frozen. The key components of contextual prompting are defined as follows:

Topic Control In the t-th turn, the dialogue state $\boldsymbol{s}_t \in \mathbb{R}^{d_s}$ is defined as the concatenation of the context embedding $\boldsymbol{u}_t$ and the preference embedding $\mathcal{P}_u$, namely $\boldsymbol{s}_t = \boldsymbol{u}_t \oplus \mathcal{P}_u$. The topic controller takes the dialogue state $\boldsymbol{s}_t$ as input and predicts the topic as:

$$\hat{\mathbf{T}} = \mathrm{softmax}(\mathbf{W}^{\mathrm{TC}}\cdot \boldsymbol{s}_t), \tag{9}$$

where $\mathbf{W}^{\mathrm{TC}} \in \mathbb{R}^{d_t\times d_s}$ is the weight matrix, and $\hat{\mathbf{T}} \in \mathbb{R}^{d_t}$ denotes the topic distribution. The predicted topic g is selected via $g = \arg\max_i(\hat{\mathbf{T}}_i)$, and then is filled in a special [MASK] token. The [MASK] token with topic information will be used as one of prompts in the later dialogue generation. We denote $\mathbf{T} \in \mathbb{R}^{d_t}$ as the ground truth vector of the topic distribution, and define the following objective to optimize the topic controller:

$$\mathcal{L}_{\text{TC}} = -\frac{1}{d_t}\sum_{i=1}^{d_t}\left[\boldsymbol{T}_i \log \hat{\boldsymbol{T}}_i + (1-\boldsymbol{T}_i)\log(1-\hat{\boldsymbol{T}}_i)\right]. \tag{10}$$

Prompt Search Existing works design dialogue prompts only based on the dialogue context. But predicted topics and user profiles can also provide additional information. Intuitively, they influence the prompt encodings by bootstrapping semantic extraction from the PLM. Therefore, we apply prompt search to enhance promptings via Transformer encoders. We first prepend a prompt sequence of m vectors as $\mathbf{S}^t = [\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots, \boldsymbol{S}_m]$. Then we encode both user inputs and the dialogue history as contextual representations, namely $\mathbf{H}^t = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_N]$, where $\boldsymbol{h}_i$ represents the embedding of the i-th token. We also define user preference matrix as $\mathbf{E}^t = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_k]$, where $\boldsymbol{e}_i$ is the knowledge representation of the i-th item in I_k. The optimization of promptings is conducted by three Transformer-based multi-head attention layers:

$$\mathbf{S}_0^t = \mathrm{MHA}(\mathbf{H}^t, \mathbf{H}^t, \mathbf{H}^t), \tag{11}$$

$$\mathbf{S}_1^t = \mathrm{MHA}(\mathbf{S}_0^t, \hat{\mathbf{T}}, \hat{\mathbf{T}}), \tag{12}$$

$$\mathbf{S}_2^t = \mathrm{MHA}(\mathbf{S}_1^t, \mathbf{E}^t, \mathbf{E}^t), \tag{13}$$

$$\mathbf{S}^t = \mathrm{FeedForward}(\mathbf{S}_2^t). \tag{14}$$

Here $\mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ stands for the multi-head attention function, which takes a query matrix $\mathbf{Q}$, a key matrix $\mathbf{K}$ and a query matrix $\mathbf{V}$ as input, and outputs the updated embedding matrix:

$$\begin{aligned}&\mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})\\ &= \underset{i\in[1,h]}{\mathrm{concat}}\left[\mathrm{Attention}_i\left(\mathbf{Q}\mathbf{W}^Q, \mathbf{K}\mathbf{W}^K, \mathbf{V}\mathbf{W}^V\right)\right]\cdot \boldsymbol{W}^O,\end{aligned} \tag{15}$$

$$\mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathbf{V} \cdot \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d}}\right). \quad (16)$$

$\mathrm{FeedForward}(\boldsymbol{x})$ denotes a two-layer fully connected network with a ReLU activation function:

$$\mathrm{FeedForward}(\boldsymbol{x}) = \mathbf{W}_2 \cdot \mathrm{ReLU}(\mathbf{W}_1\boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2, \quad (17)$$

where $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V$, $\mathbf{W}_1$, $\mathbf{W}_2$, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are model parameters. $\mathbf{S}_0^t$ is computed via self-attention on contextual embeddings. $\mathbf{S}_1^t$ and $\mathbf{S}_2^t$ are representation matrices obtained by cross-attention with topics and the user preference. Finally, $\mathbf{S}^t$ is the updated prompting matrix, which will be also used as dialogue prompts.

Prompt-Based Dialogue Generation Given the topic and prompting vectors, CP-Rec employs prompt-based dialogue generation for system replies. Assuming that the system output S_t has l tokens $(y_1, \cdots, y_l)$, we utilize GPT2 (Radford et al. 2019) to compute S_t by sampling from:

$$P(S_t) = \prod_{i=1}^{l} P\left(y_i \left| y_{<i}, \boldsymbol{h}_{1:N}, [\mathtt{MASK}], \boldsymbol{S}_{1:m}\right.\right), \quad (18)$$

where $\boldsymbol{h}_{1:N}$ and $\boldsymbol{S}_{1:m}$ are contextual and prompting vectors in $\mathbf{H}^t$ and $\mathbf{S}^t$. To better introduce the recommended items in dialogues, following Liang et al. (2021), we add a special token [ITEM] into the vocabulary. All items in the dialogue corpus are masked with the [ITEM] tokens. In the generated output, [ITEM] is replaced by the matched item. In total T turns of dialogues, the training objective of contextual prompting is to minimize the following equation:

$$\mathcal{L}_{\mathrm{prompt}} = -\sum_{t=1}^{T} \left(\log P(S_t) + \lambda \mathcal{L}_{\mathrm{TC}}\right), \quad (19)$$

where λ is a weighted hyperparameter.

Experiments

Experiment Setup

Datasets The preprocessed datasets and baselines are implemented in CRSLab (Zhou et al. 2021). We use five CRS datasets: (1) *ReDial* (Li et al. 2018) contains movie recommendation dialogues generated by Amazon Mechanical Turk workers. (2) *DuRecDial* (Liu et al. 2020) is a human-to-human CRS dataset with multi-type dialogues in various domains. (3) *TG-ReDial* (Zhou et al. 2020b) is a topic-guided CRS dataset, which focuses on natural topic transitions that lead to recommendations. (4) *OpenDialKG* (Moon et al. 2019) is a parallel corpus with dialogues and reasoning paths in KG. (5) *INSPIRED* (Hayati et al. 2020) is a sociable CRS dataset with annotated recommendation strategies. Some statistics about datasets are presented in Table 1.

Dataset	Dialogues	Utterances	Domains
ReDial	10006	182150	Movie
DuRecDial	10200	156000	Movie, Music
TG-ReDial	10000	129392	Movie
OpenDialKG	13802	91209	Movie, Book
INSPIRED	1001	35811	Movie

Table 1: Dataset statistics.

Baselines We compare our proposed model with some existing CRS baselines: (1) *ReDial* (Li et al. 2018) contains a RNN-based conversational system and an auto-encoder-based recommender system. (2) *KBRD* (Chen et al. 2019) uses DBpedia to capture semantics of contextual items for recommendation, and applies Transformer for text generation. (3) *KGSF* (Zhou et al. 2020a) applies knowledge-aware recommendations with DBpedia and ConceptNet (Speer, Chin, and Havasi 2017), and uses Transformer-based dialogue generator. (4) *TG-ReDial* (Zhou et al. 2020b) contains a BERT-based recommender system and a GPT2-based conversational system. (5) *MGCG* (Liu et al. 2020) is a GRU-based CRS towards multi-type recommendation dialogues.

In the conversation task, we further adopt the following baselines: (1) *Transformer* (Vaswani et al. 2017) applies a Transformer-based encoder-decoder framework to generate responses. (2) *GPT2* (Radford et al. 2019) is a pretrained Transformer model which is finetuned on each dataset. (3) *AutoPrompt* (Shin et al. 2020) is a discrete prompt-learning model with automatically generated templates. (4) *Prefix tuning* (Li and Liang 2021) is a prompt-learning method which searches continuous prompting tokens. AutoPrompt and prefix tuning use frozen GPT as the dialogue generators, which share the same setting with our CP-Rec.

In the recommendation task, we introduce extra baselines as follows: (1) *Popularity* ranks the items according to historical recommendation frequencies. (2) *TextCNN* (Kim 2014) is a CNN-based recommender model with textual features. (3) *GRU4Rec* (Hidasi et al. 2016) is a GRU-based recommender, which learns to recommend via the user interaction history. (4) *BERT* (Devlin et al. 2019) is an implementation of the BERT model for dialogue-based recommendations. (5) *SASRec* (Kang and McAuley 2018) uses Transformers to encode the user interaction history.

Metrics For recommendation tasks, we rank all items and calculate Hit@10, MRR@10, and NDCG@10 according to top-10 items in the ranking list. For conversation tasks, we adopt three metrics: (1) BLEU@1 measures the word overlap between the generated utterance and the ground truth. (2) Distinct@2 measures the proportion of unique 2-grams in the generated utterances. A higher Distinct means a higher diversity of responses. (3) Perplexity (PPL) is an indicator of whether the response is grammatical.

Recommendation Evaluations

Item Recommendations Table 2 presents the performance of models on the recommendation task. First, knowledge-aware recommendations are more precise than traditional ones. First, ReDial, GRU4Rec, SASRec, and TG-ReDial perform better than Popularity and TextCNN, because they consider interaction sequences and capture the user preference. Second, KBRD and KGSF achieve even better performance, since they incorporate KGs and more semantics. Finally, our model outperforms all baselines. The user profile obtained the dialogue history and exter-

Model	ReDial			DuRecDial			TG-ReDial			OpenDialKG			INSPIRED		
	Hit	MRR	NDCG	Hit	MRR	NDCG	Hit	MRR	NDCG	Hit	MRR	NDCG	Hit	MRR	NDCG
ReDial	0.015	0.005	0.007	0.046	0.013	0.021	0.004	0.001	0.001	0.006	0.003	0.004	0.058	0.012	0.022
KBRD	0.173	0.071	0.095	0.571	0.274	0.343	0.026	0.011	0.015	0.472	0.345	0.376	0.120	0.071	0.082
KGSF	0.176	0.069	0.094	0.568	0.285	0.353	0.017	0.005	0.007	0.498	0.357	0.391	0.117	0.069	0.080
TG-ReDial	0.185	0.087	0.110	0.111	0.075	0.083	0.010	0.005	0.006	0.123	0.068	0.082	0.163	0.102	0.117
Popularity	0.054	0.022	0.029	0.033	0.011	0.016	0.003	0.001	0.001	0.060	0.021	0.030	0.155	0.065	0.086
TextCNN	0.057	0.022	0.030	0.536	0.308	0.362	0.010	0.005	0.006	0.335	0.237	0.261	0.088	0.038	0.050
GRU4Rec	0.006	0.002	0.003	0.081	0.047	0.055	0.001	0.001	0.001	0.007	0.002	0.003	0.014	0.006	0.008
BERT	0.007	0.002	0.003	0.038	0.019	0.023	0.002	0.001	0.001	0.058	0.022	0.030	0.004	0.009	0.016
SASRec	0.059	0.053	0.066	0.129	0.027	0.050	0.021	0.007	0.010	0.043	0.017	0.023	0.153	0.091	0.106
CP-Rec w/o K	0.059	0.023	0.031	0.107	0.056	0.068	0.003	0.001	0.002	0.058	0.022	0.031	0.135	0.060	0.075
CP-Rec w/o D	0.142	0.071	0.089	0.543	0.260	0.326	0.025	0.008	0.012	0.435	0.260	0.298	0.162	0.088	0.103
CP-Rec	**0.229***	**0.093***	**0.123***	**0.578***	**0.364***	**0.407***	**0.030***	**0.014***	**0.018***	**0.533***	**0.397***	**0.385***	**0.186***	**0.118***	**0.156***

Table 2: Recommendation evaluation results. We compare our model with baselines on Hit@10, MRR@10, and NDCG@10. * denotes the significant improvements over the comparative methods (paired t-test, $p < 0.05$).

Model	ReDial			DuRecDial			TG-ReDial			OpenDialKG			INSPIRED		
	BLEU	Dist	PPL	BLEU	Dist	PPL	BLEU	Dist	PPL	BLEU	Dist	PPL	BLEU	Dist	PPL
ReDial	0.041	0.071	408.3	0.054	0.249	452.0	0.122	0.037	654.8	0.075	0.242	1067	0.037	0.310	1340
KBRD	0.287	0.094	61.94	0.473	0.468	39.40	0.366	0.472	73.50	0.306	0.294	94.17	0.240	0.575	215.8
KGSF	0.305	0.117	257.5	0.050	0.163	260.0	0.398	0.480	230.7	0.026	0.290	453.2	0.030	0.122	468.4
TG-ReDial	0.360	0.068	19.43	0.588	0.143	8.261	0.585	0.507	12.52	0.442	0.180	19.93	0.321	0.423	64.72
Transformer	0.208	0.085	47.52	0.416	0.501	23.11	0.352	0.389	43.87	0.227	0.201	67.67	0.191	0.289	136.5
GPT2	0.041	0.159	26.65	0.005	0.561	8.688	0.043	0.502	17.55	0.061	0.508	21.49	0.205	0.457	72.04
AutoPrompt	0.122	0.142	99.82	0.209	0.457	9.012	0.332	0.431	22.76	0.444	0.307	16.67	0.270	0.359	102.8
Prefix Tuning	0.115	0.126	105.4	0.179	0.433	9.873	0.339	0.433	22.10	0.421	0.288	19.57	0.305	0.477	57.01
CP-Rec w/o TC	0.283	0.061	100.4	0.377	0.403	15.26	0.374	0.429	37.86	0.440	0.307	22.16	0.342	0.481	38.42
CP-Rec w/o PS	0.122	0.040	833.9	0.010	0.140	378.2	0.055	0.206	478.4	0.034	0.120	893.4	0.033	0.277	877.1
CP-Rec	**0.405***	**0.176***	**8.480***	**0.602***	**0.585***	**3.990***	**0.587***	**0.510***	**2.570***	**0.526***	**0.511***	**9.280***	**0.356***	**0.577***	**20.66***

Table 3: Conversation evaluation results. We compute BLEU@1, Distinct@2, and Perplexity of each model. * denotes the significant improvements over the comparative methods (paired t-test, $p < 0.05$).

nal knowledge is more powerful in knowledge-aware recommendations. The better the model fits the datasets, the lower the PPL.

Ablation Study We adopt an ablation study to explore the effects of dialogues and knowledge on recommendation results. We fuse the contextual representations and the knowledge-based user preferences following Eq. (6) in user profiles. Two variants of CP-Rec, namely CP-Rec w/o D and CP-Rec w/o K, are implemented without incorporating contextual and knowledge-based embeddings, respectively. As illustrated in Table 2, knowledge and context both promote recommendations. In particular, we observe a more significant performance decrease of CP-Rec w/o K. We infer that the external knowledge may contain additional significant features for conversational recommendations.

Conversation Evaluations

Automatic Evaluations We report results of the conversation task in Table 3. In general, the CRS with Transformer-based dialogue generator, especially GPT2, generates higher quality conversations. ReDial uses a hierarchical RNN for dialogue generation and performs poorly among the baselines. KBRD and KGSF use knowledge-enhanced decoders and perform better than traditional Transformer in BLEU and Distinct. GPT2 and TG-ReDial have similar performance because they share the same backbone and both finetuned on datasets. We also note that prompt-based baselines, i.e., AutoPrompt and prefix tuning, achieve competitive, and even better performance with finetuned GPT2. Among these baselines, our CP-Rec outperforms finetuning counterparts and other prompt-based methods in all metrics. CP-Rec performs prompt search to integrate more semantics into GPT2, and conducts joint task planning and topic elaboration. In this way, it can be effectively context-aware, and generates coherent and informative dialogues.

Human Evaluations To further verify the dialogue quality of CP-Rec, we conduct a human evaluation on the Amazon Mechanical Turk platform. We randomly sample 100 dialogues from the test set of ReDial dataset. For each sample, we present the dialogue context and the replies of CP-Rec and baselines to three different workers without order. Each worker is asked to rate responses from 0 to 5 in terms of coherence, fluency, and informativeness. We also introduce the ground truth replies in the dataset as human responses. Table 4 presents the results of the average scores. Generally, our model performs best in all metrics, supporting the superiority of CP-Rec in generating coherent and informative responses. Our model incorporates context, topics and KGs to enhance prompts. This approach enhances the linguistic

Model	Coherence	Fluency	Informativeness
ReDial	2.94	2.86	2.44
KGSF	3.11	3.08	3.09
CP-Rec	**3.35**	**3.40**	**3.20**
Human	3.82	3.76	3.91

Table 4: Human evaluation results. We evaluate dialogue coherence, fluency and informativeness by human ratings.

Model	TG-ReDial		
	Hit@1	Hit@10	Hit@50
PMI	0.033	0.118	0.366
MGCG	0.591	0.817	0.881
TG-ReDial	0.631	0.831	0.862
CP-Rec	**0.680***	**0.842***	**0.904***

Table 5: The quality of task planning. We compute Hit@1, Hit@10, and Hit@50 of each model on the topic prediction. * denotes the significant improvements over the comparative methods (paired t-test, $p < 0.05$).

capabilities of PLMs.

Ablation Study To clarify what boosts the performance of contextual prompting, we remove topic control and prompt search, denoted as CP-Rec w/o TC and CP-Rec w/o PS. As shown in Table 3, the performance of two model variants drops dramatically. This demonstrates that the proposed components are essential for CP-Rec to handle multi-task recommendation dialogues. We further analyze the sensitivity of prompt size m to the performance. We train the CP-Rec with various numbers of prompting vectors. As illustrated in Figure 3, the size of prompting sequence has little effect on BLEU and Distinct. Increasing m does not bring significant improvement to the dialogue quality. In our experiments, we set the default value of m as 10.

The Quality of Task Planning To gain more insights, we study the quality of task planning conducted by the CP-Rec. We record the model performance in predicting topics on the TG-ReDial dataset and adopt Hit@n $(n = 1, 10, 50)$ as evaluation metrics. We compare the performance of CP-Rec with baselines applicable to multi-task recommendation dialogs, i.e., TG-ReDial and MGCG. We also measure the point-wise mutual information (PMI) with the last topic for ranking. As shown in Table 5, CP-Rec is consistently better in all evaluation metrics. The above baselines mainly use dialogue history for task planning. In contrast, we encode the external knowledge into dialogue states, which improves CP-Rec's ability to identify subtasks and lead topics.

Case Study

Table 6 shows an interactive example of movie recommendation dialogue generated by CP-Rec. We follow the utterance in ReDial dataset to interact with the system. The outputs of KGSF and CP-Rec are presented for comparison. The ground truth replies are also presented as human responses. We mark all the recommended items in italics and the planned subtasks at the beginning of responses. We note that the CP-Rec's responses are more purposeful and have a stronger recommendation orientation. Besides, CP-Rec's recommended movies are more relevant to the context and users' preferences. It shows that topic planning and elaboration work in recommendation dialogues, which maintain the conversation coherence and informativeness.

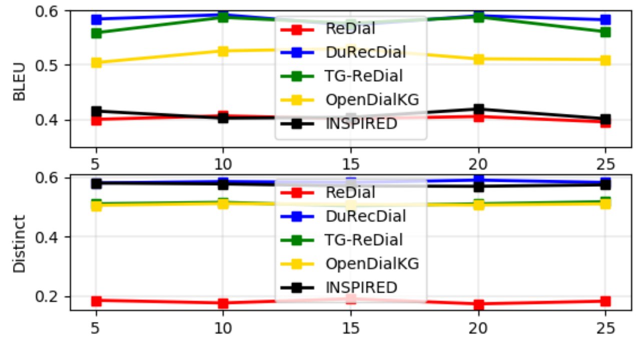

Figure 3: Sensitivity analysis of the prompt size m.

	Dialogue Sample
User:	Hi! How is your weekend?
Human:	I'm doing well. Looking for a good movie?
KGSF:	Fine. What kind of movies do you like?
CP-Rec:	[Greeting] Going well. What are you looking for?
User:	I'm looking for something scary. My favorite is *Taken*, what can you recommend?
Human:	I like *Taken*, too. If you like Liam Neeson, how about *The Commuter*? We saw that recently and enjoyed it.
KGSF:	If you like scary movie I liked *Frozen (2010)*.
CP-Rec:	[Request] I have some good ones! Have you seen *The Island (2015)* with Scarlett Johansen?
User:	Yeah, that was kind of funny. Anything else?
Human:	Did you see *The Grey*? Another Neeson film and my husband says it was good.
KGSF:	I think you would like to watch it.
CP-Rec:	[Recommend] I also like *Schindler's List* and Liam Neeson is in it.
User:	Nope, I never saw that one. Thanks so much. Have a great day!
Human:	Have a nice day.
KGSF:	I hope you enjoy them!
CP-Rec:	[Quit] I think you will enjoy it. Goodbye.

Table 6: Case study. The movie recommendation dialogues generated by the human, KGSF and CP-Rec.

Conclusion

In this paper, we develop CP-Rec, a novel conversational recommender system with contextual prompting. It conducts joint task planning and topic elaboration to generate coherent and informative dialogues. We implement knowledge-aware recommendations with external KG and propose contextual prompting for dialogue generation. Integrating topic control and prompt search, CP-Rec plans subtasks sequentially, integrates semantics comprehensively, and replies fluently. Experimental results show our CP-Rec significantly outperforms previous state-of-the-art models.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Project 62076096, Shanghai Municipal Project 20511100900, and Shanghai Knowledge Service Platform Project (No. ZF1213).

References

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901.

Cai, W.; and Chen, L. 2020. Predicting user intents and satisfaction with dialogue-based conversational recommendations. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, 33–42.

Chen, K.; and Sun, S. 2021. Knowledge-based conversational recommender systems enhanced by dialogue policy learning. In *Proceedings of the 10th International Joint Conference on Knowledge Graphs*, 10–18.

Chen, Q.; Lin, J.; Zhang, Y.; Ding, M.; Cen, Y.; Yang, H.; and Tang, J. 2019. Towards knowledge-based recommender dialog system. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1803–1813.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference on the North American Chapter of the Association for Computational Linguistics*, 4171–4186.

Fu, Z.; Xian, Y.; Zhu, Y.; Zhang, Y.; and de Melo, G. 2020. Cookie: A dataset for conversational recommendation over knowledge graphs in e-commerce. *arXiv preprint arXiv:2008.09237*.

Gao, T.; Fisch, A.; and Chen, D. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 3816–3830.

Gu, Y.; Han, X.; Liu, Z.; and Huang, M. 2022. PPT: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 8410–8423.

Hayati, S. A.; Kang, D.; Zhu, Q.; Shi, W.; and Yu, Z. 2020. Inspired: Toward sociable recommendation dialog systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 8142–8152.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations*, 1–10.

Hirano, T.; Higashinaka, R.; and Matsuo, Y. 2016. Analyzing post-dialogue comments by speakers–how do humans personalize their utterances in dialogue? In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 157–165.

Jannach, D.; and Manzoor, A. 2020. End-to-end learning for conversational recommendation: A long way to go? In *Proceedings of the 7th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems co-located with 14th ACM Conference on Recommender Systems*, 72–76.

Jiang, Z.; Xu, F. F.; Araki, J.; and Neubig, G. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8: 423–438.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *Proceedings of IEEE International Conference on Data Mining*, 197–206.

Kasahara, T.; Kawahara, D.; Tung, N.; Li, S.; Shinzato, K.; and Sato, T. 2022. Building a personalized dialogue system with prompt-tuning. In *Proceedings of the Conference on the North American Chapter of the Association for Computational Linguistics*, 96–105.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1746–1751.

Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2): 167–195.

Lei, W.; Zhang, G.; He, X.; Miao, Y.; Wang, X.; Chen, L.; and Chua, T.-S. 2020. Interactive path reasoning on graph for conversational recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2073–2083.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 3045–3059.

Li, R.; Kahou, S.; Schulz, H.; Michalski, V.; Charlin, L.; and Pal, C. 2018. Towards deep conversational recommendations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 9748–9758.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 4582–4597.

Liang, Z.; Hu, H.; Xu, C.; Miao, J.; He, Y.; Chen, Y.; Geng, X.; Liang, F.; and Jiang, D. 2021. Learning neural templates for recommender dialogue system. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 7821–7833.

Liao, L.; Takanobu, R.; Ma, Y.; Yang, X.; Huang, M.; and Chua, T.-S. 2022. Topic-guided relational conversational recommender in multi-domain. *IEEE Transactions on Knowledge and Data Engineering*, 34(5): 2485–2496.

Liu, Z.; Wang, H.; Niu, Z.-Y.; Wu, H.; Che, W.; and Liu, T. 2020. Towards conversational recommendation over multi-type dialogs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1036–1049.

Ma, W.; Takanobu, R.; and Huang, M. 2021. CR-walker: Tree-structured graph reasoning and dialog acts for conversational recommendation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1839–1851.

Madotto, A.; Lin, Z.; Winata, G. I.; and Fung, P. 2021. Few-shot bot: Prompt-based learning for dialogue systems. *arXiv preprint arXiv:2110.08118*.

Moon, S.; Shah, P.; Kumar, A.; and Subba, R. 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 845–854.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8): 9.

Ren, X.; Yin, H.; Chen, T.; Wang, H.; Hung, N. Q. V.; Huang, Z.; and Zhang, X. 2020. CRSAL: Conversational recommender systems with adversarial learning. *ACM Transactions on Information Systems*, 38(4): 1–40.

Rendle, S. 2010. Factorization machines. In *Proceedings of IEEE International Conference on Data Mining*, 995–1000.

Sarkar, R.; Goswami, K.; Arcan, M.; and McCrae, J. P. 2020. Suggest me a movie for tonight: Leveraging knowledge graphs for conversational recommendation. In *Proceedings of the 28th International Conference on Computational Linguistics*, 4179–4189.

Sarwar, B.; Karypis, G.; Konstan, J.; and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, 285–295.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of European Semantic Web Conference*, 593–607.

Shin, T.; Razeghi, Y.; Logan IV, R. L.; Wallace, E.; and Singh, S. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 4222–4235.

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4444–4451.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, 5998–6008.

Wang, X.; Zhou, K.; Wen, J.-R.; and Zhao, W. X. 2022. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1–9.

Xu, H.; Moon, S.; Liu, H.; Liu, B.; Shah, P.; and Philip, S. Y. 2020. User memory reasoning for conversational recommendation. In *Proceedings of the 28th International Conference on Computational Linguistics*, 5288–5308.

Zhou, K.; Wang, X.; Zhou, Y.; Shang, C.; Cheng, Y.; Zhao, W. X.; Li, Y.; and Wen, J.-R. 2021. CRSLab: An open-source toolkit for building conversational recommender system. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 185–193.

Zhou, K.; Zhao, W. X.; Bian, S.; Zhou, Y.; Wen, J.-R.; and Yu, J. 2020a. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1006–1014.

Zhou, K.; Zhou, Y.; Zhao, W. X.; Wang, X.; and Wen, J.-R. 2020b. Towards topic-guided conversational recommender system. In *Proceedings of the 28th International Conference on Computational Linguistics*, 4128–4139.