Customized Conversational Recommender Systems

Customized Conversational Recommender Systems

Shuokai Li[1,2,⋆], Yongchun Zhu[1,2,⋆], Ruobing Xie[3], Zhenwei Tang[4], Zhao Zhang[1], Fuzhen Zhuang[5,6] ✉, Qing He[1,2,⋆] ✉, and Hui Xiong[7]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] WeChat Search Application Department, Tencent, China
[4] King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
[5] Institute of Artificial Intelligence, Beihang University, Beijing 100191, China
[6] SKLSDE, School of Computer Science, Beihang University, Beijing 100191, China
[7] Artificial Intelligence Thrust, The Hong Kong University of Science and Technology, Guangzhou, China
{lishuokai18z,zhuyongchun18s,zhangzhao2021,heqing}@ict.ac.cn
ruobingxie@tencent.com zhenwei.tang@kaust.edu.sa
zhuangfuzhen@buaa.edu.cn xionghui@ust.hk

Abstract. Conversational recommender systems (CRS) aim to capture user's current intentions and provide recommendations through real-time multi-turn conversational interactions. As a human-machine interactive system, it is essential for CRS to improve the user experience. However, most CRS methods neglect the importance of user experience. In this paper, we propose two key points for CRS to improve the user experience: (1) *Speaking like a human*, human can speak with different styles according to the current dialogue context. (2) *Identifying fine-grained intentions*, even for the same utterance, different users have diverse fine-grained intentions, which are related to users' inherent preference. Based on the observations, we propose a novel CRS model, coined **C**ustomized **C**onversational **R**ecommender **S**ystem (CCRS), which customizes CRS model for users from three perspectives. For human-like dialogue services, we propose multi-style dialogue response generator which selects context-aware speaking style for utterance generation. To provide personalized recommendations, we extract user's current fine-grained intentions from dialogue context with the guidance of user's inherent preferences. Finally, to customize the model parameters for each user, we train the model from the meta-learning perspective. Extensive experiments and a series of analyses have shown the superiority of our CCRS on both the recommendation and dialogue services.

Keywords: Conversational Recommendation · Knowledge Graph · Customization · Meta Learning.

⋆ The authors are at the Key Lab of Intelligent Information Processing of Chinese Academy of Sciences. ✉ indicates corresponding authors.

1   Introduction

Recently, conversational recommender systems (CRS) [31,24,16,2,33,19,17], which capture the user current preference and recommend high-quality items through real-time dialog interactions, have become an emerging research topic. They [16,2,33] mainly require a dialogue system and a recommender system. The dialogue system elicits the user intentions through conversational interaction and responses to the user with reasonable utterances. On the other hand, the recommender system provides high-quality recommendations by user's intentions and inherent preferences. CRS has not only a high research value but also a broad application prospect [7], such as "Siri", "Cortana" etc.
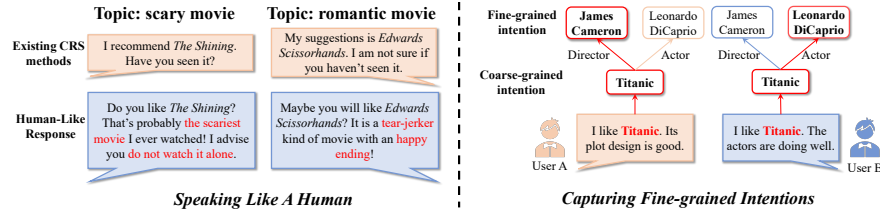
Fig. 1. An motivating example of our CCRS. The left part shows the various human speaking styles on different topics. The right part presents different users have the same coarse-grained intentions but with various fine-grained intentions.

As a kind of human-machine interactive system, improving user experience is of vital importance. However, existing CRS methods neglect the importance of user experience. Some methods [31,24] not only require lots of labor to construct rules or templates but also make results rely on the pre-processing, which hurt user experience as the constrained interaction [7]. Some other approaches [16,2,34,33,20] generate inflexible and fixed-style responses which could make users uncomfortable. Besides, some methods [2,33] only identify coarse-grained intentions and cannot provide customized recommendations. In this paper, to improve the user experience in CRS, we propose two key points:

- **Speaking like a human:** Facing various dialogue scenes, people's responses may be diverse largely in terms of speaking styles. Figure 1 (left) shows an example when the topics are about horror films and romantic movies. Obviously, a human-like dialogue system is expected to: (1) generate utterances that fit the current content semantics and topics, rather than using a fixed template; (2) generate vivid and attractive conversations, rather than short, dull and boring expressions. In this way, the user experience would be improved and user engagement would also increase, which helps identify the user intentions more accurately.
- **Identifying fine-grained intentions:** The same utterance from different users could reflect diverse fine-grained intentions. For example, in Figure 1 (right), if users mentioned the movie "*Titanic*" during the conversation, they

would have the same coarse-grained intentions to find movies related to "*Titanic*". Nevertheless, they may have different fine-grained intentions. This is because of the diversity of the inherent preferences: some users prefer movies of the actor "*Leonardo DiCaprio*", while others prefer movies of the director "*James Cameron*". Thus, modeling the fine-grained intentions which are related to the users' inherent preferences helps provide high-quality recommendations.

Along this line, we design a novel model **C**ustomized **C**onversational **R**ecommender **S**ystem (CCRS), which customizes CRS model from three perspectives. Firstly, for the recommender service, given user mentioned entities, the key idea is to highlight user fine-grained intentions with the guidance of user's inherent preferences (i.e. the preferences on different relations of entity). Then, for the dialogue service, we generate customized utterances with the guidance of content semantics. In detail, multiple styles are modeled as style embeddings, and we aggregate multiple style embeddings into the customized speaking style embedding according to the dialogue context. Finally, we further customize the (recommendation and generation) models for each user, with the advantage of meta-learning. As user fine-grained intentions and speaking styles are sparse in CRS, we choose Model-Agnostic Meta-Learning (MAML) algorithm [6], which can rapidly learn customized model parameters.

To summarize, the contributions of this paper are as follows:

- We propose a novel customized conversational recommender system CCRS, which consists of customized recommendation and dialog services. We further customize the model parameters for each user with the advantage of meta-learning.
- We model user fine-grained intentions on entity relations, and propose multi-style generation to provide human-like dialogue service, which improves user experience.
- Extensive quantitative and qualitative experiments demonstrate that the proposed approach can significantly outperform baseline methods.

2   Method

In this section, we present our novel method to provide customized services for users, coined **C**ustomized **C**onversational **R**ecommender **S**ystem (CCRS). First, we extract the fine-grained intentions of users with the guidance of user inherent preferences. Then, we design the multi-style embeddings and generate personalized responses based on the extracted fine-grained user intentions. Finally, we adopt meta training to learn customized model parameters for each user rapidly. The overview illustration of the proposed model is presented in Figure 2.

2.1   Preliminary and Formulations

Recommender Module in CRS Given a user $u \in \mathcal{U}$ with his identifier u_{id} and his mentioned entities $e \in \mathcal{E}$, a recommender system aims to retrieve a subset of items that meet the customized user needs. To be noticed, the entities consist of item entities and non-item entities. For example, in a movie recommender

system, the item entities denote the movies and the non-item entities can be the actor/actress, director, and genre. To better model user mentioned entities, external knowledge graphs $\mathcal{G}$ are often incorporated.

Specifically, the knowledge graph $\mathcal{G}$ consists of triples (e, r, e') where the entities $e, e' \in \mathcal{E}$ and the entity relation $r \in \mathcal{R}$. $\mathcal{E}$ and $\mathcal{R}$ denote the entity set and entity relation set, respectively. Following [2,33], we utilize the knowledge graph from DBpedia [15] to learn the entity representations. As the original graph consists of redundant information, we collect all the entities appearing in the dialogue corpus and extract the subgraph following [2,33].
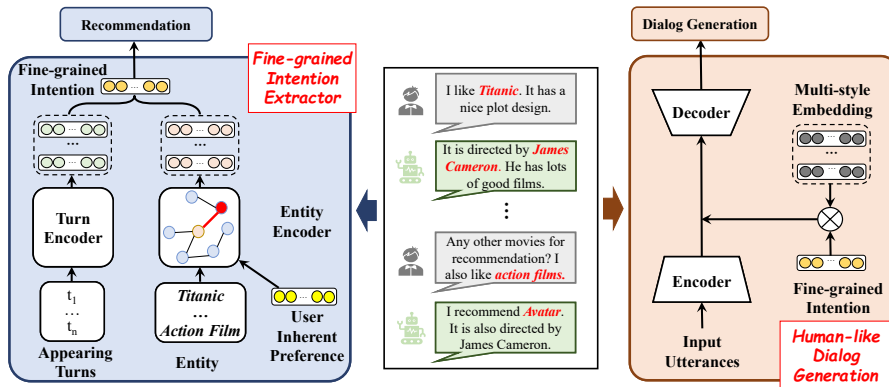
Fig. 2. The overview of our CCRS, which consists of a customized recommender part and a personalized dialogue generation part. Moreover, we use MAML to learn customized model parameters for both recommendation and dialogue generation modules.

Dialogue Module in CRS The dialogue system is designed to generate proper utterance responses in natural languages. At the T-th turn of the conversation, the dialogue system receives the dialogue history $\mathcal{H} = \{s_t\}_{t=1}^{T-1}$ and user mentioned entities $\{e_1, e_2, ..., e_n\}$. Here the entities are extracted from the dialogue utterances using entity linking. For simplify, the items are replaced by a special token "$<$unk$>$". When generating "$<$unk$>$", it means to provide recommendations for users. Then the items are provided by the recommendation part.

2.2   Fine-grained User Intentions Extraction

In this section, we extract the fine-grained user intentions by considering the entity relations and the appearing turns of the entities. These two factors are leveraged to learn the importance of user mentioned entities.

Entity Encoder Given the user mentioned entities and corresponding knowledge graph, previous methods [2,33] mainly leverage Relational Graph Convolutional Networks (R-GCNs) [23] to incorporate the structural information and learn

entity representations. Though R-GCNs keep a distinct linear projection for each type of entity relation, the projections are fixed for different users, regardless of the fine-grained user intentions. Indeed, the fixed projections could only model the coarse-grained preferences (i.e. *knowing what entities do users like*), and ignore the customized fine-grained user intentions on entity relations (i.e. *not knowing what types of entity relations do users like*).

Motivated by [11], we proposed to capture the fine-grained user intentions by modeling personalized preference on entity relations. For each user, with the guidance of the user inherent embedding, we learn the customized attention weights on various relations and aggregated the neighboring entities according to the customized attention weights.

Given the triple (e, r, e') in knowledge graph, where e is the user mentioned entity, $e' \in N(e)$ is the neighbour of e and r is the relation between e and e', our goal is to learn the fine-grained user attentions on relation r. First, to learn the fine-grained entity information, we project the l-th layer's entity embeddings ($h^{l-1}(e) \in \mathbb{R}^{d \times 1}$) into multi-head representations:

$$T_i^{l-1}(e) = W_i^T h^{l-1}(e), \qquad S_i^{l-1}(e') = W_i^S h^{l-1}(e'), \qquad (1)$$

where $W_i^T \in \mathbb{R}^{\frac{d}{k} \times d}$ and $W_i^S \in \mathbb{R}^{\frac{d}{k} \times d}$ are trainable weights, and $h^0(e) = \text{Emb}_e(e)$ is initialized randomly. Next, as different users may have various inherent preferences, we keep a distinct relation-aware matrix A_r^i and use $\gamma_{<r,u>}^i$ to represent the user intentions on distinct entity relation r. Concretely, $\gamma_{<r,u>}^i = \text{Vec}(A_r^i) W_u \boldsymbol{U}_{id}$ is the similarity between A_r^i and the inherent user embedding $\boldsymbol{U}_{id}$, here $\text{Vec}(A_r^i)$ represents flattening the matrix into a vector. Then the fine-grained user intentions on the relation r for i-th head is calculated as:

$$g_i(r, u) = S_i^{l-1}(e')^{\mathbf{T}} A_r^i T_i^{l-1}(e) \cdot \frac{\gamma_{<r,u>}^i}{\sqrt{d}}. \qquad (2)$$

Finally, the overall user fine-grained intention on entity relation r is calculated by concatenating k heads together:

$$G(r, u) = \underset{N(e)}{\text{Softmax}}(\text{Concat}(g_1, ..., g_k))[e']. \qquad (3)$$

Here the "Softmax" is the normalization of the source nodes, and "$[e']$" denotes the e'-th element. By this, user fine-grained intentions on relations $G(r, u)$ are learned with the guidance of user inherent preferences $\boldsymbol{U}_{id}$.

Similar to the attention calculation procedure, we model the information of source nodes message using k heads projections and concatenated the multi-head information like Eq 3:

$$f_i(e') = M_r^i W_i^M h^{l-1}(e'), \qquad F(e') = \text{Concat}(f_1, ..., f_k), \qquad (4)$$

where $M_r^i \in \mathbb{R}^{\frac{d}{k} \times \frac{d}{k}}$ and $W_i^M \in \mathbb{R}^{\frac{d}{k} \times d}$ denote the i-th head message matrices.

Then we aggregate the message from the source nodes to the target node with the guidance of user inherent attention $G(r, u)$, which finally leads to personalized entity representations and captures the customized and fine-grained user intentions:

$$h_*^l(e) = \sum_{\forall e' \in N(e)} (G(r, u) \cdot F(e')). \tag{5}$$

Finally, the entity embedding is finally updated by residual connection [9]:

$$h^l(e) = \sigma(W^A h_*^l(e)) + h^{l-1}(e), \tag{6}$$

where $W^A \in \mathbb{R}^{d \times d}$ is the aggregation matrix and $\sigma(\cdot)$ denotes the activation functions (in practice, we use GELU [10]). In the following, we utilize the last layer's (layer L) representation $h^L(e)$ as the entity representation and denote $h^L(e)$ as $h(e)$ for simplification.

Now given the user mentioned entities $\{e_1, e_2, ..., e_n\}$ through the conversations, we encode them into fine-grained entity representations $H_u = (h(e_1), h(e_2), ... , h(e_n))$, where $h(e_j) \in \mathbb{R}^d$ denotes the entity embedding of e_j.

Turn Encoder By the entity encoder, we learn the fine-grained user intentions H_u on mentioned entities. However, these entities are not equally important. In CRS, the importance of entities is also influenced by the temporal factor. That is, the entities that appeared in the later turns are prone to be more important than early entities. Motivated by the position embedding technique [25,4] in NLP, we take the appearing turns of entities into consideration:

$$\boldsymbol{\mu}_u^o = \text{Attn}(O_u) = \text{Softmax}(\boldsymbol{w}_2^O \text{Tanh}(W_1^O O_u)), \tag{7}$$

where $o_i = \text{Emb}_t(t_i)$ is the turn embedding and $O_u = (o_1, ..., o_n)$ is the combination of o_i. Here t_i is a scalar, which denotes the appearing turn of the entity e_i, and Emb_t is the turn embedding layer. We then use the turn importance (i.e. $\boldsymbol{\mu}_u^o$) of entities to better learn fine-grained user intentions.

Fine-grained Intention Encoder Actually, the importance of entities is also influenced by the entities themselves. Thus we calculate the self-importance of entities like Eq. 7: $\boldsymbol{\mu}_u^r = \text{Attn}(H_u)$. Finally, we calculate the user representations $\boldsymbol{p}_u$ in terms of entity and turn importance, and recommend items according to user intentions:

$$\boldsymbol{p}_u = \frac{1}{2}(\boldsymbol{\mu}_u^r + \boldsymbol{\mu}_u^o)H_u, \qquad \boldsymbol{p}_{rec}(i) = \text{Softmax}(\boldsymbol{p}_u \widetilde{H})[i], \tag{8}$$

where $\widetilde{H}$ is the embedding matrix of the whole items and i is the index of items.

2.3   Customized Dialogue Generation

Sequence-to-Sequence Model The seq2seq framework has been verified in NLP and recommendation [1,27,28]. It consists of an encoder that encodes the input utterances into high-level representations and a decoder that generates the responses. Following [2,33], we leverage the Transformer [25] as base architecture.

Given the input utterance $x = (x_1, ..., x_{n_c})$ with dialogue history, the encoder extracts information from x. Then the decoder receives the encoder outputs and generates a representation $\boldsymbol{q}$ at each decoding time step. According to the representation $\boldsymbol{q}$, the generator calculates a probability distribution over the whole vocabulary to determine the generated tokens.

Multi-Style Generation Actually, the speaking style depends on the current user intention. That is to say, when talking about horror films and romance movies, the speaking style varies definitely. Thus we would like to model the speaking styles of users and perform customized generation.

First, we pre-define n_s *latent speaking style embeddings* $\mathrm{L} = \{\boldsymbol{l}_1, ..., \boldsymbol{l}_{n_s}\} \in \mathbb{R}^{d \times n_s}$. Then for each user, the corresponding styles vary according to current dialogue contexts and user fine-grained intention. Thus the multi-style vocabulary bias $\boldsymbol{g}_u$ is learned by proper speaking styles:

$$\boldsymbol{\mu}_u^m = \boldsymbol{p}_u W^C \mathrm{L}, \qquad \boldsymbol{g}_u = \boldsymbol{\mu}_u^m \mathrm{L}^{\mathbf{T}}, \tag{9}$$

where $\boldsymbol{p}_u$ is the user fine-grained intentions learned by Eq. 8 and W^C is the similarity matrix. The selected style embeddings fit the context semantics and user inherent preferences. Finally, we add the vocabulary bias to the original generator to perform personalized utterances generation:

$$\boldsymbol{p}_{dial} = \mathrm{softmax}(W^G \boldsymbol{q} + \mathcal{F}(\boldsymbol{g}_u) + \boldsymbol{b}), \tag{10}$$

where $\mathcal{F} : \mathbb{R}^d \rightarrow \mathbb{R}^{|V|}$ maps the vocabulary bias vector into a $|V|$-dimension vector, which consists of two fully connected layers.

2.4 Customized Model Training

In previous sections, we describe the details of our model and capture the customized user preferences from the *design of network* perspective. In this section, we will model the customized user preferences from the *training* perspective.

Training Loss A common way to train the parameters is the back-propagation algorithm. For the recommendation part, we could leverage a common cross-entropy loss to train the recommendation part:

$$\mathcal{L}_{rec} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{N_u} \sum_{n=1}^{N_u} \log \boldsymbol{p}_{rec}(y_{u,n}), \tag{11}$$

where y_{un} denotes the actual preference of user u and N_u is the number of the movies in which the user u is interested.

For the dialogue generation module, the common training loss is also the cross-entropy loss:

$$\mathcal{L}_{dial} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{N_u^g} \sum_{n=1}^{N_u^g} \log \boldsymbol{p}(s_{n,u} | s_{(n-1),u}, ..., s_{1,u}, x_{n,u}), \tag{12}$$

where N_u^g denotes the whole utterance of user u, and $s_{n,u}$ is for the gold response tokens.

Nevertheless, it lacks the personality for various users and could not tune the network according to the customized and fine-grained user preferences. Motivated by [14], we adopt the meta-learning framework [6] to learn the communal user preferences and customized user preferences for a specific user.

Algorithm 1 The training algorithm of recommendation and dialogue generation parts.

Require:
  The training model: $m = \text{rec}$ or dial,
  β_m and ν_m: inner and outer learning rates.
  Randomly initialized the inner θ_m^{inner} and outer θ_m^{outer} params
  while not converge **do**
      Sample batch of users $\mathcal{U} \sim \mathcal{D}_{tr}$
      for user u in $\mathcal{U}$ **do**
          $(\mathcal{D}_u^{sup}, \mathcal{D}_u^{qu}) \sim \mathcal{D}_{tr,u}$
          $\mathcal{L}_1(u) = \mathcal{L}_{m,\mathcal{D}_u^{sup}}(f_{\theta_m^{\text{inner}}, \theta_m^{\text{outer}}})$
          Inner update: $\phi(u)_m^{\text{inner}} \leftarrow \theta_m^{\text{inner}} - \beta_m \nabla_{\theta_m^{\text{inner}}} \mathcal{L}_1(u)$
          $\mathcal{L}_2(u) = \mathcal{L}_{m,\mathcal{D}_u^{qu}}(f_{\phi(u)_m^{\text{inner}}, \theta_m^{\text{outer}}})$
      end for
      Global update
      $\theta_m^{\text{outer}} \leftarrow \theta_m^{\text{outer}} - \nu_m \sum_{u \in \mathcal{U}} (\nabla_{\theta_m^{\text{outer}}} \mathcal{L}_2(u) + \nabla_{\theta_m^{\text{outer}}} \mathcal{L}_1(u))$
      $\theta_m^{\text{inner}} \leftarrow \theta_m^{\text{inner}} - \nu_m \sum_{u \in \mathcal{U}} (\nabla_{\theta_m^{\text{inner}}} \mathcal{L}_2(u) + \nabla_{\theta_m^{\text{inner}}} \mathcal{L}_1(u))$
  end while

Meta Training The meta training includes the inner update and global update. We first define predicting each user's preference as an individual task and sample a set of records as support set $\mathcal{D}_u^{sup}$, while others as the query set $\mathcal{D}_u^{qu}$ for each user. In the inner update, the model updates the inner parameters θ^{inner} (see Section Parameters Setting for details) to learn customized user tastes, according to the user's unique item-consumptions (i.e., the support set). It takes a single gradient step with inner learning rate β:

$$\phi(u)^{\text{inner}} \leftarrow \theta^{\text{inner}} - \beta \nabla_\theta \mathcal{L}_{\mathcal{D}_u^{sup}}(\mathcal{M}), \tag{13}$$

where $\mathcal{M}$ denotes the training parameters of the whole model. As the inner parameters are updated in different directions for different users, it captures the user customized preferences. The following procedure is global update and its goal is to learn a communal parameters initialization, such that each of the users' customized preferences would be met from the common initialization with a few update steps, i.e., capturing the customized preferences rapidly:

$$\theta \leftarrow \theta - \nu \sum_{u \in \widetilde{U}} \nabla_\theta \mathcal{L}_{\mathcal{D}_u^{qu}}(\mathcal{M}(\phi(u)^{\text{inner}})), \tag{14}$$

where θ includes the whole parameters of $\mathcal{M}$, and ν is the outer learning rate.

In practice, we first train the recommender part to learn the entity embeddings and take the customized recommendation. When the recommender part converges, the dialogue generation module is optimized with the guidance of user intentions on entities, and personalized utterances are generated. The detailed training algorithm of recommendation and dialogue generation is shown in Algorithm 1.

Parameters Setting For the recommendation part, the inner parameters are set to $\theta_{\text{rec}}^{\text{outer}} = \{\text{Emb}_u, \text{Emb}_e\}$, and the outer parameters are $\theta_{\text{rec}}^{\text{inner}} = \theta_{\text{rec}} \setminus \theta_{\text{rec}}^{\text{outer}}$, where $\theta_{\text{rec}} = \{\theta | \theta \in \mathcal{M}_{\text{rec}}\}$. For the dialogue generation part, The parameters are also divided into two categories: the inner parameters $\theta_{\text{dial}}^{\text{inner}} = \{\text{Encoder}', W_j^L |_{j=1}^{n_s}\}$ and the outer parameters $\theta_{\text{dial}}^{\text{outer}} = \theta_{\text{dial}} \setminus \theta_{\text{dial}}^{\text{inner}}$, where $\theta_{\text{dial}} = \{\theta | \theta \in \mathcal{M}_{\text{dial}}\}$.

3 Experiment

In this section, we first introduce the details of our experiments and then answer the following questions: **RQ1**: How does our CCRS perform on recommendation and dialogue generation compared with the SOTA baselines? **RQ2**: Does our CCRS capture the fine-grained user intentions? **RQ3**: Does our CCRS generate human-like responses? **RQ4**: How do different components of CCRS benefit the performances?

3.1 Experimental Setup

Dataset. To evaluate the effectiveness of CCRS, we conduct experiments on real-world dataset ReDial. It contains 10,006 conversations and 182,150 utterances. The total number of users and movies are 956 and 51699, respectively.

For meta-learning, we define each task as a user's interactive conversations. Then we group the train, validation, and test sets by the user id, and the ratio of samples is about 8:1:1. For each user, half of the user conversations are used as the query set, and the remaining conversations are used as the support set. During the meta test phase, CCRS is fine-tuned on the support set $\mathcal{D}_{test}^{sup}$ and tested on the query set $\mathcal{D}_{test}^{qu}$. For a fair comparison with baseline methods, when training these models, we add the meta test support set $\mathcal{D}_{test}^{sup}$ into the training data.

Baselines. We consider the following baselines: (1) *Popularity* ranks the items according to the historical recommendation frequencies. (2) *TextCNN* [12]: encodes utterances to extract user intent by CNN-based model. (3) *ReDial* [16] is a CRS method which adopts an auto-encoder for recommendation. (4) *KBRD* [2] adopts the external knowledge graph DBpedia to enhance the user representations. (5) *KGSF* [33] incorporates both the semantic KG ConceptNet and the entity KG DBpedia for modeling user preferences. (6) *KECRS* [30] constructs a high-quality KG and it proposes the Bag-of-Entity loss and the infusion loss to better integrate KG with CRS for generation. (7) *RevCore* [20] collects user reviews on movies to enhance the recommendation and dialogue generation modules.

Evaluation Metrics. For the recommendation task, we evaluate whether it recommends high-quality items. So we adopt Recall@k (following [33]), MRR@k and

NDCG@k (following [34]) for evaluation ($k = 10, 50$). For the dialogue generation task, we evaluate the performance by the automatic and human evaluations. In the automatic evaluations, we adopt BLEU [22] and F1 to estimate the generation quality, and Distinct n-gram ($n = 3, 4$) to measure the diversity at sentence level. In the human evaluations, we invite three annotators to score whether the generations are fluent (*Fluency*) and plenty of useful information (*Informativeness*). The range of scores is 0 to 2, and the final result is the average scores of the three annotators. **Implementation Details.** We implement our approach based on PyTorch framework. For the recommendation part, the entity embedding size is set to 128. We choose the number of entity relation extractor layers $L = 1$ and the number of heads $k = 4$. The network parameters are initialized by glorot uniform [8]. For training, the inner and outer learning rates are set to 0.006 and 0.003, respectively. For the dialogue part, the dimension of word embeddings is set to 300, and the number of styles n_s equals 4. For training, the inner and outer learning rates are set to 0.0003 and 0.001, respectively. For both parts, we use Adam [13] optimizer with default parameter setting, and gradient clipping restricts the gradients within [0, 0.1].

3.2   Overall Performance (RQ1)

Recommendation. The recommendation results on ReDial are shown in Table 1. From Table 1, we have the following observations. (1) First, our CCRS outperforms CRS baselines (ReDial, KBRD, KGSF, KECRS, and RevCore) by a large margin, which shows the superiority of the fine-grained user intentions extraction and meta training framework. With the guidance of inherent user preferences, the fine-grained user intentions are modeled, and MAML further customizes user preferences by the local update. Moreover, though CCRS does not consider ConceptNet and external user reviews, it outperforms KGSF and RevCore significantly, which also shows the effectiveness of our customization recommendation module. (2) Second, CCRS beats the non-CRS method TextCNN. The reason is that TextCNN directly learns user representation by the whole history, which suffers from the sparsity and noise of the utterances. (3) Finally, KBRD, KGSF, and KECRS perform better than ReDial, which proves the effectiveness of incorporating external knowledge graphs. Besides, KGSF outperforms KBRD, as KGSF leverages an extra word-oriented KG ConceptNet.

Dialogue Generation. We also evaluate our CCRS on the dialogue generation task and the main results are listed in Table 2. From Table 2, we have the following observations. (1) First, our CCRS outperforms the baselines significantly. The improvement on BLEU shows that the generation of CCRS is more consistent with the ground truth, and the high value of Distinct n-gram reflects the diversity of CCRS's results. (2) Then, from the human evaluation perspective, CCRS also generates the most fluent and informative responses, which are more human-like responses. The main reason is that CCRS considers multi-style generations according to the current content semantics and inherent user preferences. The human-like generations then improve the user experience, and users are more

Table 1. The recommendation results. The marker * indicates that the improvement is statistically significant compared with the best baseline (t-test with p-value < 0.05).

Dataset	ReDial					
Method	HR@10	HR@50	MRR@10	MRR@50	NDCG@10	NDCG@50
Population	6.47	17.68	0.0158	0.0204	0.0346	0.0617
Text CNN	6.57	16.51	0.0235	0.0275	0.0425	0.0661
ReDial	11.79	30.11	0.0551	0.0628	0.0895	0.1273
KBRD	15.20	33.26	0.0593	0.0677	0.0979	0.1382
KGSF	17.00	35.72	0.0637	0.0717	0.1072	0.1497
KECRS	11.57	31.95	0.0573	0.0639	0.0911	0.1291
RevCore	17.36	36.66	0.0659	0.0741	0.1103	0.1528
CCRS	**19.09***	**38.41***	**0.0717***	**0.0808***	**0.1193***	**0.1618***
w/o TE	17.26	36.58	0.0661	0.0751	0.1092	0.1520
w/o RE&TE	16.49	35.23	0.0633	0.0720	0.1063	0.1463

Table 2. Evaluation results on dialogue generation. Flu. and Inf. stand for Fluency and Informativeness, respectively. The marker * indicates that the improvement is statistically significant compared with the best baseline (t-test with p-value < 0.05).

Dataset	ReDial						
Method	BLEU	F1	Dist-2	Dist-3	Dist-4	Flu.	Inf.
ReDial	1.213	0.183	0.089	0.393	0.798	0.83	0.96
KBRD	1.287	0.192	0.118	0.571	1.212	0.95	1.03
KGSF	1.629	0.227	0.123	0.647	1.583	1.23	1.32
KECRS	1.088	0.125	0.078	0.351	0.761	0.85	0.99
RevCore	1.236	0.186	0.105	0.553	1.321	1.21	1.33
CCRS	**2.386***	**0.267***	**0.146***	**0.776***	**1.924***	**1.36***	**1.43***
Only-Meta	2.129	0.258	0.131	0.692	1.772	1.30	1.40

willing to chat with CCRS. (3) Besides, compared with ReDial, we can see that the external knowledge graph also contributes to the generation.

3.3 Qualitative Results on Recommendation (RQ2)

Fine-grained user intentions on entity relations. In this part, we present qualitative examples to show that we capture fine-grained customized user intentions.

We first choose three movies "The Matrix", "Titanic" and "Sleepless in Seattle" and three different relations "Director", "Actor" and "Producer". Then we randomly pick up four users to show their fine-grained intentions on relations. The results are shown in Figure 3 left. (1) Firstly, we can see that the user's preferences on relations vary. For example, user 1 pays more attention to the "Director" relation, while user 4 cares about "Actor". So it is necessary to learn the user attention on the entity

(a) Fine-grained user intentions   (b) Turn Importance
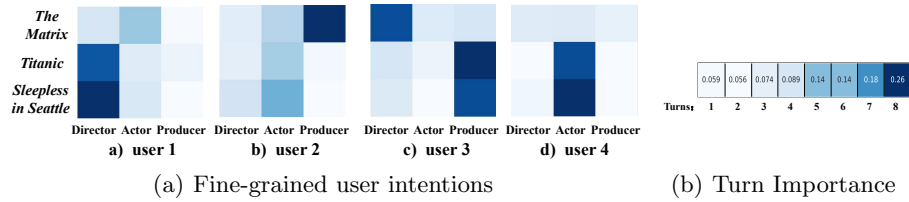
Fig. 3. The visualization of fine-grained user intentions on different relations (left), and the importance of the dialogue turns (right).

relation, and we aggregate more information from the "Director" relation for user 1, while the "Actor" relation for user 4. (2) Secondly, even for a fixed user, the intentions on different movies also vary. For user 1, he likes the *director* of "Sleepless in Seattle" and the *actor* of "The Matrix". (3) Last but not least, in the original corpus, user 4 mentioned that he loves Tom Hanks, which is the star of "Sleepless in Seattle", and he also likes the movie "Castaway" which also includes Tom Hanks. These observations show that our CCRS correctly captures the fine-grained user intentions on entity relations and provides high-quality recommendations.

Turn Encoder. For the turn encoder, we first conduct an experiment to prove that the appearing turn of the entities contributes to the recommendation performances. Given a user's mentioned entities, we try to rank them purely according to the appearing turn. Then we use the first and last entity's first-order neighbors as the candidate pool. If the candidate pool includes the target movie, this recommendation is viewed as correct. After the calculation, we find the accuracy of the first entity is 2.32%, while the last entity is 3.50%. This confirms that the entities near the current turn of the dialogue are more important for learning user preference. Secondly, we visualize the importance of each turn learned by CCRS in Figure 3 right. We observe that with the increase of turns, the attention weights become larger, which helps to recommend high-quality items according to the customized user preferences.

3.4   Qualitative Results on Dialogue (RQ3)

In this part, we present some qualitative examples to illustrate the personalized generation. Firstly, as shown in Table 3, when talking about different topics of movies, our CCRS could generate multi-style responses according to the current content semantics. While the speaking style of KGSF is monotonous. Thus our CCRS improves the user experience, and the user would feel like chatting with a real person.

Then, we visualize the style distribution vector $\boldsymbol{\mu}_u^m$ (see Eq. 9) in Figure 4, which controls the choice of the four speaking style embeddings. On the one hand, for different types of movies, the style distribution vector varies significantly. For example, for horror film, $\boldsymbol{\mu}_u^m$ is prone to choose Style 3, while the romantic movie is Style 1. On the other hand, for the same type of movie, the style distribution is similar among different users, but not identical. In a word, CCRS chooses customized style vectors for various kinds of movie topics, which leads to human-like generations.

Table 3. The human-like responses for different types of movies.

Topic 1: Horror film	User: I recently watched *The Shining* and it was great. KGSF: Maybe you will also like *Scream*. CCRS: I'm not sure if you have seen *The Conjuring*. I would say it's pretty horror. It scared me for whatever reason.
Topic 2: Romantic Movie	User: Hello! Can you suggest some movies like *Roman Holiday*? KGSF: *50 First Date* is a good choice. CCRS: What about *The Notebook*? The characters suffer lots of hardships and get an happy-ending. It tells a moving story!
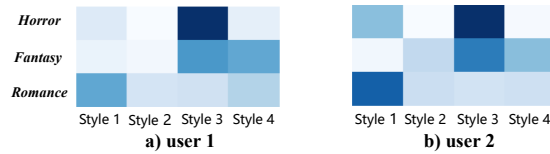
Fig. 4. The probability $\boldsymbol{\mu}_u^m$ of choosing the different styles. We can see that for different type of movies, the style probability varies.

3.5 Ablation Study (RQ4)

Recommendation. In our recommendation part, we propose to learn the fine-grained user intentions, temporal factors of entities, and the meta-learning framework to achieve the customized recommendations. Here we would like to examine the effectiveness of each part and show the results in Table 1. For meta training, CCRS w/o RE&TE denotes training KBRD from the meta-learning perspective (KBRD leverages RGCN to learn the entity representations). CCRS w/o RE&TE beats KBRD, which shows the effectiveness of meta training. For the remaining two parts, CCRS w/o TE denotes training CCRS with only the entity relation encoder, and it outperforms without entity relation encoder (i.e. CCRS w/o RE&TE). CCRS incorporates both the entity relation encoder and the turn encoder, and it beats without turn encoder (i.e. CCRS w/o TE) significantly. These results show the usefulness of each part of CCRS. With the help of the fine-grained user intention, the appearing turns of entities, and meta-learning, CCRS achieves good performances.

Dialogue Generation. In our generation part, we adopt the multi-style generator and the meta training to generate personalized utterances. We also conduct ablation studies to examine the effectiveness of each part. First, the Only-Meta denotes training KBRD with MAML. We can see that MAML significantly improves the quality (BLEU) and diversity (Distinct n-gram) simultaneously. This is not surprising, as we update the network parameters in the user specific direction in the inner update, which leads to the customized generations. Then, CCRS beats Only-Meta, which shows the effectiveness of the multi-style generator. Compared with the Only-Meta, though the improvements on BLEU and F1 scores are marginal, CCRS generates more personalized responses.

4   Related Work

Conversational Recommender Systems Conversational Recommender Systems (CRS) [31,24,16,2,33,19,17] aim to provide high-quality items through the interactive conversations. It mainly consists of a recommender system that identifies the user customized preferences given the item-consumption history and a dialogue system that converses with the users and collects their preferences. Early conversational recommender systems [3,24] mainly collects the user preference via the pre-defined questions. They pay more attention to recommendation accuracy, and the dialogue system is an auxiliary part, which is implemented by simple or pre-defined patterns. Recently, several works focus on building end-to-end CRS models. [16] proposed a standard CRS dataset named ReDial and an end-to-end model which consists of a hierarchical recurrent encoder, a switching decoder, an RNN-based sentiment analysis module, and an autoencoder-based recommendation module. Moreover, [2,33] proposed to incorporate the external knowledge to improve the CRS performances. [2] mainly leverages the knowledge graph (KG) of user mentioned entities, which includes movies, actors, etc., while [33] incorporates both word-oriented and entity-oriented KGs, and adopts Mutual Information Maximization to align word-level and entity-level representations. Other works focus on selecting proper inter-active action (policy) during the conversations. [19] constructs a goal sequence which includes question answering, chit-chat, and recommendation phrase. Then they characterized the goal planning process and focus type switch during conversations. [34] focused on topic-guided CRS and proposed a topic prediction model. Furthermore, [32] proposed an open-source CRS toolkit CRSLab, which provides a unified and extensible framework for previous CRS works. Moreover, [20] proposes a review-enhanced framework, in which user reviews are incorporated to enhance the CRS performances. [18] further learns templates automatically for utterances generation.

Meta-Learning Meta-learning is also named learning to learn, which aims to improve new tasks' performance with several similar tasks. The work can be grouped into three clusters, e.g. metric-based [26], model-based [37], and optimization-based [6] approaches. Recently, meta-learning has also been applied to recommender systems [21,14,36,38,35,29] and natural language processing [5]. In this paper, we leverage meta-learning to learn customized conversational recommendation model, and provide personalized service, which is different from previous works. To the best of our knowledge, this is the first attempt in adapting meta-learning to conversational recommendation for model customization and improving user experience.

5   Conclusion

In this paper, we proposed a novel approach CCRS, which aims to improve the user experience and explore the customization in CRS from three perspectives. For the customized recommendation results, we capture fine-grained intentions by exploring user inherent preference on entity relations and the appearing turn of the entities. For the customized dialogue interactions, we proposed a multi-style

generator, which generates responses in customized speaking styles. Finally, for the model training, we adopted the meta-learning framework to enhance the recommendation and dialogue generation via customized model parameters. The extensive experiments show that our CCRS outperforms competitive baselines.

Acknowledgments

References

1. Bahdanau, D., Cho, K.H., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR (2015)
2. Chen, Q., Lin, J., Zhang, Y., Ding, M., Cen, Y., Yang, H., Tang, J.: Towards knowledge-based recommender dialog system. In: EMNLP. pp. 1803–1813 (2019)
3. Christakopoulou, K., Radlinski, F., Hofmann, K.: Towards conversational recommender systems. In: SIGKDD. pp. 815–824 (2016)
4. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL. pp. 4171–4186 (2019)
5. Dong, B., Yao, Y., Xie, R., Gao, T., Han, X., Liu, Z., Lin, F., Lin, L., Sun, M.: Meta-information guided meta-learning for few-shot relation classification. In: COLING. pp. 1594–1605 (2020)
6. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017)
7. Gao, C., Lei, W., He, X., de Rijke, M., Chua, T.S.: Advances and challenges in conversational recommender systems: A survey. AI Open **2**, 100–126 (2021)
8. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 249–256. JMLR Proceedings (2010)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
10. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
11. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: WWW. pp. 2704–2710 (2020)
12. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP 2014. p. 1746–1751 (2014)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv (2014)
14. Lee, H., Im, J., Jang, S., Cho, H., Chung, S.: Melu: Meta-learned user preference estimator for cold-start recommendation. In: SIGKDD. pp. 1073–1082 (2019)
15. Lehmann, J., Isele, R., Jakob, M., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web **6**(2), 167–195 (2015)
16. Li, R., Ebrahimi Kahou, S., Schulz, H., Michalski, V., Charlin, L., Pal, C.: Towards deep conversational recommendations. NeurIPS **31**, 9725–9735 (2018)
17. Li, S., Xie, R., Zhu, Y., Ao, X., Zhuang, F., He, Q.: User-centric conversational recommendation with multi-aspect user modeling. In: SIGIR (2022)

18. Liang, Z., Hu, H., Xu, C., et al.: Learning neural templates for recommender dialogue system. In: EMNLP. pp. 7821–7833 (2021)
19. Liu, Z., Wang, H., Niu, Z.Y., Wu, H., Che, W., Liu, T.: Towards conversational recommendation over multi-type dialogs. In: ACL. pp. 1036–1049 (2020)
20. Lu, Y., Bao, J., Song, Y., Ma, Z., Cui, S., Wu, Y., He, X.: Revcore: Review-augmented conversational recommendation. arXiv preprint arXiv:2106.00957 (2021)
21. Pan, F., Li, S., Ao, X., Tang, P., He, Q.: Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In: SIGIR. pp. 695–704 (2019)
22. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL. pp. 311–318 (2002)
23. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference. pp. 593–607. Springer (2018)
24. Sun, Y., Zhang, Y.: Conversational recommender system. In: SIGIR. pp. 235–244 (2018)
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 5998–6008 (2017)
26. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. NeurIPS **29**, 3630–3638 (2016)
27. Wu, Y., Xie, R., Zhu, Y., Zhuang, F., Ao, X., Zhang, X., Lin, L., He, Q.: Selective fairness in recommendation via prompts. In: SIGIR (2022)
28. Wu, Y., Xie, R., Zhu, Y., Zhuang, F., Zhang, X., Lin, L., He, Q.: Personalized prompts for sequential recommendation. arXiv (2022)
29. Xie, R., Wang, Y., Wang, R., Lu, Y., Zou, Y., Xia, F., Lin, L.: Long short-term temporal meta-learning in online recommendation. In: WSDM (2022)
30. Zhang, T., Liu, Y., Zhong, P., et al.: Kecrs: Towards knowledge-enriched conversational recommendation system. arXiv preprint arXiv:2105.08261 (2021)
31. Zhang, Y., Chen, X., Ai, Q., Yang, L., Croft, W.B.: Towards conversational search and recommendation: System ask, user respond. In: CIKM. pp. 177–186 (2018)
32. Zhou, K., Wang, X., Zhou, Y., Shang, C., Cheng, Y., Zhao, W.X., Li, Y., Wen, J.R.: Crslab: An open-source toolkit for building conversational recommender system. In: ACL: System Demonstrations. pp. 185–193 (2021)
33. Zhou, K., Zhao, W.X., Bian, S., Zhou, Y., Wen, J.R., Yu, J.: Improving conversational recommender systems via knowledge graph based semantic fusion. In: SIGKDD. pp. 1006–1014 (2020)
34. Zhou, K., Zhou, Y., Zhao, W.X., Wang, X., Wen, J.R.: Towards topic-guided conversational recommender system. In: COLING. pp. 4128–4139 (2020)
35. Zhu, Y., Ge, K., Zhuang, F., Xie, R., Xi, D., Zhang, X., Lin, L., He, Q.: Transfer-meta framework for cross-domain recommendation to cold-start users. In: SIGIR. pp. 1813–1817 (2021)
36. Zhu, Y., Liu, Y., Xie, R., Zhuang, F., Hao, X., Ge, K., Zhang, X., Lin, L., Cao, J.: Learning to expand audience via meta hybrid experts and critics for recommendation and advertising. In: KDD. pp. 4005–4013 (2021)
37. Zhu, Y., Tang, Z., Liu, Y., Zhuang, F., Xie, R., Zhang, X., Lin, L., He, Q.: Personalized transfer of user preferences for cross-domain recommendation. In: WSDM. pp. 1507–1515 (2022)
38. Zhu, Y., Xie, R., Zhuang, F., Ge, K., Sun, Y., Zhang, X., Lin, L., Cao, J.: Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In: SIGIR. pp. 1167–1176 (2021)