Improving Conversational Recommender System Via Contextual and Time-Aware Modeling With Less Domain-Specific Knowledge

Lingzhi Wang, Shafiq Joty, Wei Gao, Xingshan Zeng, and Kam-Fai Wong, Senior Member, IEEE

Abstract—Conversational Recommender Systems (CRS) has become an emerging research topic seeking to perform recommendations through interactive conversations, which generally consist of generation and recommendation modules. Prior work on CRS tends to incorporate more external and domain-specific knowledge like item reviews to enhance performance. Despite the fact that the collection and annotation of the external domain-specific information needs much human effort and degenerates the generalizability, too much extra knowledge introduces more difficulty to balance among them. Therefore, we propose to fully discover and extract the internal knowledge from the context. We capture both entity-level and contextual-level representations to jointly model user preferences for the recommendation, where a time-aware attention is designed to emphasize the recently appeared items in entity-level representations. We further use the pre-trained BART to initialize the generation module to alleviate the data scarcity and enhance the context modeling. In addition to conducting experiments on a popular dataset (ReDial), we also include a multi-domain dataset (OpenDialKG) to show the effectiveness of our model. Experiments on both datasets show that our model achieves better performance on most evaluation metrics with less external knowledge and generalizes well to other domains. Additional analyses on the recommendation and generation tasks demonstrate the effectiveness of our model in different scenarios.

Index Terms—Conversational recommendation, pre-trained language model, recommender system.

I. INTRODUCTION

Conversational Recommender Systems (CRS) [1], [2], [3], [4] have recently attracted many researchers due to the booming of e-commerce platforms. A CRS aims to provide high-quality recommendations to users through conversations. Different from the traditional recommender systems, it focuses on learning users’ preferences through natural language interaction with users, and has a high impact in e-commerce.

Manuscript received 10 May 2022; revised 29 August 2023; accepted 21 April 2024. Date of publication 7 May 2024; date of current version 27 September 2024. This work was supported in part by the CUHK direct under Grant 4055209 and in part by the CUHK Knowledge Transfer Project Fund under Grant KPF23GWP20. Recommended for acceptance by P. Bogdanov. (Corresponding author: Lingzhi Wang.)

Lingzhi Wang and Kam-Fai Wong are with The Chinese University of Hong Kong, Hong Kong (e-mail: lwang@se.cuhk.edu.hk; kf-wong@se.cuhk.edu.hk).
Shafiq Joty is with the Nanyang Technological University, Singapore 639798 (e-mail: sjoty@ntu.edu.sg).
Wei Gao is with the Singapore Management University, Singapore 188065 (e-mail: weigao@smu.edu.sg).
Xingshan Zeng is with the Huawei Noah’s Ark Lab, Hong Kong (e-mail: zeng.xingshan@huawei.com).
Digital Object Identifier 10.1109/TKDE.2024.3397321

Table I: Two Conversation Examples from ReDial (The Upper Row) and OpenDialKG (The Lower Row) Datasets

Seeker: Can you recommend some newer released family friendly movies?
Recommender: Yes you should watch Power Rangers (2017), Captain Underpants: The First Epic Movie was great too
Seeker: Well that is a good suggestion, but my kids don’t like Power Rangers (2017) much.
Recommender: How old are they?
Seeker: They are 14 years old and 10 years old.
Recommender: Despicable Me (2010) and Wonder Woman (2017) are great flicks

Seeker: Can you recommend a good book by Virginia Woolf?
Recommender: Sure! Have you read the To The Lighthouse?
Seeker: No I have not. When was it released?
Recommender: It came out in 1927.
Seeker: Do you know any other books she wrote?
Recommender: She wrote many. You may be interested in The Waves or Mrs. Dalloway.

Seeker is a user who asks movie (or book) recommendation and Recommender is supposed to do chit-chat and recommendation. The mentioned items (i.e., movie and book) are in blue.
The collection and annotation of the external knowledge need much human effort. Some of the collected external knowledge may be domain-specific and lack generalizability when facing broader application scenarios, e.g., the item reviews introduced by [4] (some recommended items like sports or unpopular items might lack reviews). On the other hand, the previous work largely ignores the timing or sequence order information by regarding the appeared items in context as a set rather than a sequence, which is contrary to the goal of the system. As one of the goals of CRS is to guide the users to express their preferences explicitly through conversations, we believe that the timing or sequence information would be a crucial factor to further improve the performance of a CRS.

Therefore, instead of exploring more domain-specific external knowledge to assist the learning of user preferences, we choose to fully discover and extract the internal knowledge from the context and propose a time-aware user preference modeling. Concretely, we capture both entity-level and contextual-level representations to model the user preferences. The entity-level representations summarize user preferences with the appeared items in context by our proposed time-aware modeling. As for the contextual-level representations extracted by a context encoder, they reflect the semantic- and discourse-level user preferences, which cannot be captured by entity-level ones. These two representations complement each other to enhance the recommendation results.

To elaborate the motivation of our recommendation module design, Table I shows two conversation examples from CRS datasets, where the upper conversation focuses on recommending movies and the lower one is about recommending books. It is observed that the context contains rich and key information. For example, “my kids don’t like Power Rangers (2017)” in the first example of Table I indicates the user shows negative opinion to Power Rangers (2017) and the CRS is supposed to not recommend items that are similar to the negative item. Only focusing on the appeared items (e.g., Power Rangers (2017)) like the previous work [1], [2], [3], [4] would misunderstand the users’ true preference. In addition to the potential misunderstanding of the user intention, there is also rich and direct information that is vital for a recommendation. For example, “Do you know any other books she wrote.” in the second example of Table I implies that the user needs a specific book recommendation from the aforementioned author Virginia Woolf. Such kind of context information provides precise user preference, which should not be ignored by a CRS. Therefore, in addition to the entity-level representation, it’s essential to consider the contextual-level user interest representation.

As for the generation module, most of the previous methods [1], [2], [3], [4] employ a general encoder-decoder framework and train the model from scratch. It suffers from the overfitting issue on the relatively small dataset, as most of the CRS datasets only contain about 10k conversations and most of the syntactic structure of the utterances is simple and boring. Therefore, we use the pre-trained BART [6] model to initialize our generation module and thus alleviate the data scarcity effect in capturing meaningful context features. Furthermore, the utilization of pretrained language models (PLMs) can also benefit the learning of contextual-level user interests compared to other non-pretrained conversation modeling modules (e.g., GRU and Transformer), thus further improving contextual-level recommendations.

Different from most previous works [1], [2], [3], [4], [7] that only test the effectiveness of their models on a single domain data, we conduct experiments on two public and popular CRS datasets ReDial [1] and OpenDialKG [5] under the multi-domain scenario. The results show that our model can achieve better recommendation and generation performance when assisted with less domain-specific external knowledge, and validate that our model generalizes well to other domains. Further analyses also demonstrate the effectiveness of our proposed modules.

The main contributions of this work can be summarized as follows:

- We propose to combine both entity-level and contextual-level representations for modeling user preference in conversational recommendation, which achieves better performance with less domain-specific external knowledge compared to the previous works and generalizes well to other domains.
- We point out the limitation of the previous entity modeling method and contribute a time-aware user preference modeling method to enhance the recommendation.
- We adopt the pre-trained language model BART to enhance the semantic learning and diversity of our generated responses.

The remainder of this paper is organized as follows. The related work is surveyed in Section II. Section III presents the proposed approach. And Sections IV and V present the experimental setup and the corresponding results respectively. Finally, conclusions are drawn in Section VI.

II. RELATED WORK

In this section, we provide an in-depth review of the related research work from three different aspects, conversational recommender system, traditional recommender system and pre-trained languages model.

A. Conversational Recommender System

Conversational Recommender System (CRS) has attracted many researchers’ interest in recent years. CRS can be broadly classified as a context-aware recommender system (CARS) [8], [9], [10], wherein contextual information is integrated to offer more refined personalized recommendations. For CRS, Various task formulations with different hypotheses and application

1 Code is publicly available at https://github.com/Lingzhi-WANG/CTA-CRS.
scenarios have been proposed. We summarize them into three categories and introduce them in detail as below.

1) **Question Driven Systems:** As the rating or click feedback in traditional recommender systems is limited in that they do not exactly tell why the users like or dislike an item, the feedback from the users could be very sparse. Therefore, question driven systems are proposed to effectively understand users’ preferences and improve the recommendations over time by asking clarifying questions. [11] proposes question-based video recommender system. [12] builds systems based on aspect-centered questions. [13] formulates the task of asking clarifying questions in open-domain information-seeking conversational systems. More recent works focus on asking attribute-central questions and developing reinforcement learning based approaches [14], [15], [16] or graph based approaches [17], [18], [19], [20].

2) **Strategies Learning in Multi-Turn CRS:** Some works in this category focus on balancing the trade-off between exploration (i.e., asking questions) and exploitation (i.e., making recommendations), especially for the cold-start users. They study the trade-off strategies to achieve engaging and successful recommendations. Some of them [21], [22], [23], [24] leverage bandit online recommendation methods and focus on cold-start scenarios, while others work on strategically asking clarification questions with fewer turns [14], [18], [25].

3) **Open-Ended CRS:** An open-ended CRS aims to make recommendation in a more natural and casual way compared with the task-oriented CRS. Many datasets have been collected or built to push forward the research of CRS, including ReDial [1], TG-ReDial (Chinese) [26], GoRecDial [27], DuRecDial (Chinese) [28], INSPIRED [29] and OpenDialKG [5] datasets. Most of them consist of around 10,000 conversations that are focused on recommendation and chit-chat on different domains. For example, ReDial is about movie recommendation, while OpenDialKG is concerned with several domains, including movie, book, sports and music. The follow-up studies on the ReDial dataset generally divide the CRS into recommendation and generation modules. For the recommendation module, the previous works tend to apply more and more external knowledge to improve the recommendation performance, e.g., entity-level knowledge graph [2], word-level knowledge graph [3] and item reviews [4]. However, it’s challenging to manage so much external knowledge via an end-to-end model. What’s more, some (like item reviews) might need much effort to collect and annotate, and are not generally applicable for all kinds of domains (e.g., some items might lack reviews). For generation module, most of the previous works adopt encoder-decoder framework and train the generation model from scratch. However, it’s difficult to learn diverse and valuable patterns from relatively small datasets. Our work further explores approaches and improves quality of CRS for this category.

**B. Recommender System**

Traditional recommender system is different from the conversational recommender system, as it generally does not focus on dialogue interactions with users. It makes more efforts on the information filtering techniques to handle information overflow problems in recent years.

The problems in the traditional recommender system can be divided into two categories [30]: recommendation with explicit feedback (i.e., the ratings given to products) and recommendation with implicit feedback (purchase history or click-through history) [31]. The former problem can be evaluated directly by differences between ground truths and predictions while the evaluation of the latter problem is usually formulated as Top-N ranking problems.

The methods proposed in the conventional recommender system can be divided into three categories: collaborative filtering based recommendation [32], content-based recommendation [33] and the hybrid of them [34]. Collaborative filtering has been the most popular recommendation technique in recent years. It hypothesizes that people who have had similar interests before would also prefer similarly in the future [35]. Many methods are proposed to identify the similarity of users or items, including memory-based methods [36], clustering methods [37], Bayesian models [38] and matrix factorization methods [39]. However, collaborative filtering methods may perform poorly when handling new items that lack history information. As for content-based recommendation methods, they usually first compare the similarity between the descriptive characteristics of items and the user profiles, and then recommend items that are similar to what users like before. Therefore, the content-based methods are usually efficient in recommending new items. But they are limited in terms of diversity and novelty [40]. The hybrid of collaborative filtering and content-based methods take advantages of both of them [34].

**C. Pre-Trained Language Model**

The pre-trained language models [6], [41], [42], [43] become more and more popular in recent years. The idea behind them is first pre-training the models on large-scale unlabeled corpus (which can be easily crawled on the Internet) and then apply them to many downstream tasks (e.g., QA system, dialogue system, summarization and machine translation). The architecture of these pre-trained language models has been developed from shallow to deep (the parameter scale is advancing from million-level [41], [43] to trillion-level [44]) along with the improvement of computational capability and the enhancement of training mechanisms. Pre-training tasks such as nearby word prediction [45], next word prediction [46], [47], masked language model [41] are widely explored in language modeling. These tasks do not need any annotated labels, so they can be trained on a huge amount of unlabeled data from the Internet, e.g., Wiki and Reddit.

We introduce how these pre-trained language models are applied to the downstream tasks. The development of utilizing pre-trained language models can be divided into two stages, embedding-based approaches and finetuning-based approaches. In embedding-based approaches, the word-, sentence- or paragraph-level embeddings from the pre-trained language models is utilized as features in the downstream tasks. As for finetuning-based approaches, they benefit the knowledge
transferring between the pre-trained language models and the
downstream tasks. A common finetuning procedure would fix
(or apply smaller learning rate to) the original parameters of the
pre-trained language model and add some finetunable adaptation
modules for the downstream tasks. Besides, prompt tuning is
getting more attention in all of finetuning methods. Discrete [48],
[49] or continuous [50] prompts based finetuning are proposed to
help bridge the gap between the pre-training and finetuning and
reduce the computational cost. Different from the domain-
specific knowledge (e.g., review of items) introduced by the
previous work [4], [51], the information learned by pre-trained
language models is general and not task-specific. The finetuning
methods based on pre-trained language model can be well gen-
eralized to other domains while the training scheme based on
domain-specific knowledge limits the model generalizability.

III. METHODOLOGY

In this section, we first formulate the task of CRS in Section
III-A, followed by the description of our generation module
in Section III-B, which is finetuned with a pre-trained sequence-
to-sequence framework BART [6]. Then we introduce how we
do recommendation based on conversation context and entity-
level information in Section III-C. Finally, we describe how we
integrate the above two modules and produce the final responses
in Section III-D.

A. Problem Formulation

A classical architecture of conversational recommender sys-
tem generally consists of two modules, named generation mod-
ule and recommendation module. It takes the history context of
a conversation $C = (t_1, t_2, \ldots, t_m)$ as input, where $t_i$ ($1 \leq i \leq m$)
is an utterance from the seeker (i.e., the user) or the CRS itself and $m$ is the number of context utterances. Each utterance $t_i$
contains word tokens $t_i = (w_{i1}, w_{i2}, \ldots, w_{in_i})$, where $n_i$
is the number of tokens in utterance $t_i$. The CRS uses its rec-
ommendation module to recommend items $I_t$ from a candidate
item set $I$ based on the context $C$ and some related external
knowledge (e.g., knowledge graphs or item reviews). Then it
embeds them into a response $R = (y_1, y_2, \ldots, y_n)$, a sequence
of $n$ tokens generated by the generation module based on the
context $C$ to produce natural responses with recommendations.

B. BART-Based Response Generation

Our response generation module follows a general Trans-
former [52] sequence-to-sequence framework. We first briefly
introduce the Transformer framework followed by the BART
pre-trained language model. Then we explain how we finetune
the BART model to enable generating responses with recom-
ended items.

Transformer: A Transformer layer typically consists of two
kinds of sub-layers: multi-head attention layers and fully con-
ected feed-forward networks. In a multi-head attention layer,
for each head $j$, the output vector is computed with the following
equation:

$$\text{Attn}_j(Q, K, V) = \text{softmax} \left( \frac{Q_j K_j^T}{\sqrt{d_k}} \right) V_j \quad (1)$$

where $Q, K, V$ are the matrices of queries, keys and values,
$d_k$ is the dimension of keys, and $Q_j = QW_j^Q$, $K_j = KW_j^K$, $V_j = VW_j^V$ are project matrices for head $j$. The output of
each head is concatenated and then mapped to the final output
field. For the fully connected feed-forward network (FFN), it
consists of two linear transformations with an ReLU activation in between:

$$\text{FFN}(X) = \text{ReLU}(XW_1 + b_1)W_2 + b_2 \quad (2)$$

A Transformer encoder layer is composed of a multi-head self-attention layer (takes the same input as $Q, K, V$) and a
feed-forward network, while a Transformer decoder layer addi-
tionally attends to the encoder output (as $K$ and $V$) for cross
attention (see the bottom part of Fig. 1).

In summary, the whole process for a general Transformer
sequence-to-sequence framework is as follows:

$$H^C = \text{Transformer}_{\text{Encoder}}(x) \quad (3)$$

$$y_k = \text{Transformer}_{\text{Decoder}}(y_{<k}, H^C) \quad (4)$$

$$L_{\text{gen}} = \sum_{k=1}^{n} - \log(p(y_k|y_{<k}, H^C)) \quad (5)$$

where $x$ is the input token sequence, and $y_{<k}$ represents the
target tokens before $y_k$. The Transformer_{Encoder} includes a
word embedding layer and several Transformer encoder layers
while the Transformer_{Decoder} is similar but additionally
includes an output projection layer. The model is learned to
minimize the negative log-likelihood of the target sequence as
in (5).

BART: BART [6] is a pre-trained language model based on
Transformer trained with several denoising objectives on
large-scale books and Wikipedia data, i.e., recovering the noisy
input processed by token masking, token deletion, text infilling,
sentence permutation and document rotation. These denoising
objectives enable the model to learn contextual representations
with unlabeled data, which can be crawled easily from the
Internet, and then used for multiple downstream tasks. BART
has been shown to be effective in many generation tasks, in-
cluding abstractive QA, summarization, machine translation at
the sentence and document level, and persona-based response
generation.

Finetune BART for CRS: As most of the available CRS
datasets are relatively small and contain only around 10K con-
dversations [1], [5], it is difficult to learn complex semantic
and discourse level dependencies only based on the training
corpus. To relieve the burden and enhance context modeling,
we choose to finetune a pre-trained BART [6] model for our
response generation module. Though the introduction of pre-
trained BART model can be regarded as a way of including more
external knowledge to the CRS, there are two major differences
between the domain-specific external knowledge (i.e., review
of items) and the general external knowledge (i.e., pre-trained
language model): (i) Difficulties in applying the knowledge. The
collection of domain-specific knowledge is time-consuming and how to design efficient networks to include the domain-specific knowledge with other knowledge becomes a new challenge, while the finetuning scheme based on the pre-trained language model is easier and can follow some regular solutions. (ii) Generalizability. The pre-trained language models can be employed in most of the tasks while the domain-specific knowledge may only exist in a few tasks or be hard to collect for some tasks.

To enable the BART model to generate the item-aware responses, we extend its original vocabulary \( \mathcal{V} \) with the item set \( \mathcal{I} \) to be \( \mathcal{V} = \mathcal{V} \cup \mathcal{I} \) (in this case, we do not tokenize the item names in an utterance and recognize them as single tokens), and then use a CRS training corpus to finetune the model. Specifically, we concatenate the utterances \( t_i \) in context \( C \) with an appended \( \langle EOT \rangle \) token in their chronological order as the input (i.e., \( x = \{t_1; \langle EOT \rangle; t_2; \ldots; \langle EOT \rangle; t_m\} \)), and maximize the probability of the ground truth response \( R \) (i.e., minimize the objective in (5)).

Finally, an integration operation is added to the generation output to make the generation aware of the recommendation. We will discuss it later in Section III-D.

C. Context-Time-Aware Recommendation

To fully understand user preferences over items from a given context \( C \), we propose to extract two kinds of information for the recommendation. The first is entity-level information, where we extract the mentioned entities (including the items in the item set \( I \)) from \( C \) and apply them to an external relational knowledge graph to perform entity linking [53]. Then the learned entity representations are summarized by a proposed time-aware attention to produce entity-level user representation. The second is contextual information produced by BART model, which is expected to capture information from the perspectives of semantic and conversational discourse. We describe the details in the following.

1) Entity-Level Representation: Entity-level representation is captured and represented by extracted items as well as their related entities (e.g., if the items are the movies, the related entities can be the authors, directors, etc.) from the observed context \( C \). Because of the data scarcity, it is not promising to summarize the extracted entities merely based on the training corpus [2]. We follow previous work and employ a relational knowledge graph (e.g., DBpedia) to enhance entity modeling.

Entity Linking: Specifically, we denote a triplet in the relational knowledge graph with \( \langle e_1, r, e_2 \rangle \), where \( e_1, e_2 \in \mathcal{E} \) are entities from the entity set \( \mathcal{E} \) and \( r \) is a kind of entity relation from the relation set \( \mathcal{R} \). For the ReDial dataset, we utilize the linked knowledge graph, a subset of DBpedia, which is released by [2]. For the OpenDialKG dataset, we also extract a subset of DBpedia by matching each item in the item set to entities in DBpedia and additionally linking them on dialogue contents by following [53].

We use an R-GCN [54] framework to encode relation-aware entity representations. The intuition behind this is that the features of an entity can be indicated and summarized by its neighboring nodes in the knowledge graph. An R-GCN computes the embeddings of the entities with multi-edge encoding with several iteration layers. Formally, the representation of an entity \( e \) at the \((l+1)\)th layer is:

\[
    h_e^{(l+1)} = \text{ReLU} \left( \sum_{r \in \mathcal{R}} \sum_{e' \in \mathcal{E}_r} \frac{1}{\deg_r} W_{e,r}^{(l)} h_{e'}^{(l)} \right)
\]

where \( h_{e'}^{(l)} \) represents the representation of entity \( e' \) at the \( l \)th layer, and \( \mathcal{E}_r \) denotes the set of neighboring nodes of \( e \) under the relation \( r \). \( \mathcal{R} = \mathcal{R} \cup \{ r_{rel} \} \) contains all the relations including

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
self loop. $W^{(t)}_{e,r}$ is a learnable relation-specific transformation matrix and $Z_{e,r}$ is a normalization factor. Therefore, for each node in the knowledge graph, it receives and aggregates the messages from its neighboring nodes after relation-specific transformation. Then it combines the information with the hidden representations to form its updated representation at the next layer. Finally, structural and relational information is encoded into the entity representation $h^E_e$ for each $e \in \mathcal{E}$ at the last layer $L$. For simplicity, we represent the representations in the final layer $L$ as $h_e$ by omitting the superscript “$L$”.

**User Preference Summarization:** Given a context $C$, we presume we can summarize the preference of the seeker with the appeared entities. We extract the appeared entities (as user preference) $\mathcal{T}_u = \{e_1, e_2, \ldots, e_{|\mathcal{T}_u|}\}$ from two perspectives: item entities (i.e., entities that appear in item set $I$) and other relevant contextual entities (mentioned in utterances but not an item in $\mathcal{T}$, e.g., an actor of a movie item. We denote them as text entities). For items that are not covered by $\mathcal{E}$, we ignore them by following the previous work [2], [3], [4]. The entities $e_i \in \mathcal{E}$ are sorted in the order of appearance. After looking up the entities in $\mathcal{T}_u$, from $H = \{h_e\}_{e \in \mathcal{E}}$, we get $(h_1, h_2, \ldots, h_{|\mathcal{T}_u|})$.

To summarize the entity-level user representation, previous work mainly depends on a self-attention mechanism [3], [4], where a learnable matrix is used to learn and derive each entity’s importance. Such a naive mechanism might be sub-optimal for the following two reasons: (i) no supervised signals are used to guide the model to learn knowledge about entity importance which makes it difficult to summarize accurately, especially when the training corpus is limited; (ii) it ignores the order of appearance which might affect the conversation trend (e.g., the appeared entities $A, B, C$ and $C, B, A$ probably lead to different next items in a conversation, but we will get the same summarized representation with such a mechanism). Therefore, we propose Time-aware Attention to address the limitations, where the entity-level user representation $h^E$ is calculated as follows:

$$h^E = \sum_{i=1}^{|\mathcal{T}_u|} \frac{\lambda^{i-1}}{\sum_{i=1}^{|\mathcal{T}_u|} \lambda^{i-1}} h_i$$

(7)

where $\lambda$ is the hyper-parameter to control the recency effect. The value of $\lambda$ is usually larger than 1, which means that the recently appeared items will contribute more to the next item prediction. The intuition behind the time-aware attention can be twofold: (i) The currently available conversational recommendation datasets are quite small and the previous proposed self-attention based item representation could not show promising performance (see discussion in Section V-A-1). Our simple but effective modeling mechanism can handle multi-domain datasets regardless of their sizes. (ii) One of the goals of conversational recommender system is to guide the users to better express their preferences explicitly through the ongoing conversations. Therefore, it’s naturally to put more weights on the most recently appeared items. We will discuss the effect of the value of $\lambda$ in Section V-E-1.

Finally, the time-aware entity-level recommendation probability can be computed as follows:

$$p_e = \text{softmax}(\text{mask}(h^E H^T))$$

(8)

where mask is an operation that sets all non-item entities to $-\infty$ (to make the recommendation focus on the candidate items in item set $I$), and $p_e \in \mathbb{R}^{C}$. $p_e$ will be used for recommendation together with contextual-level representation. We introduce the latter one in the next subsection.

2) **Contextual-Level Representation:** Though the time-aware entity-level representation can model the user preference from the perspective of the mentioned items in context $C$, it may lead to misunderstanding of user preferences since it ignores the content from the context. For example, if a user says “I do not like A!” We cannot capture such a negative opinion towards “A” through entity-level representation alone.

To partially address the problem and to incorporate more semantic- and discourse-level context for recommendation, we further use the context representation $H^C$ computed in (3) from generation module to yield semantic-aware prediction. Specifically, we average the context representations over the sequence as contextual-level representation for $C$ and put it through an MLP layer to give the prediction:

$$p_c = \text{softmax} \left(\text{MLP} \left(\frac{1}{|C|} \sum_{j=1}^{|C|} h^C_j\right)\right)$$

(9)

where $h^C_j$ denotes the representation for the $j$th token in the context representation $H^C$, $|C|$ is the total length of context, and $p_c \in \mathbb{R}^{C}$ reflects the effects from the content for recommendation, which can be a complement for entity-level representation. We next describe how we combine these two kinds of representations.

3) **Combination of the Entity- and Contextual-Level Recommendation:** Different from the previous work, our work considers both entity-level representation and contextual-level representation for recommendation. The final recommendation based on the above two components is defined as:

$$p_{rec} = \mu \cdot p_e + (1 - \mu) \cdot p_c$$

(10)

where $\mu (0 \leq \mu \leq 1)$ is a hyper-parameter to balance between the two kinds of recommendations. Therefore, the learning objective for the recommendation module is:

$$L_{rec} = - \sum_{i=1}^M \log p_{rec}(r_i) + \log p_{rec}(r_i)$$

(11)

where $M$ is the number of items extracted from the training corpus that need to be recommended, while $r_i \in I$ is the target item in the $i$th recommendation. $p_{rec}(r_i)$ and $p_e(r_i)$ are the corresponding prediction probabilities of the target item from the entity-level and contextual-level recommendation components, respectively.

**D. Module Integration**

So far, the responses generated by the generation module (introduced in Section III-B) are not aware of the recommendation
(introduced in Section III-C) results. We introduce an integration mechanism to incorporate the recommendation module’s results and guide the generation module to generate responses that are consistent with the user’s preference. Some of the previous works [3], [4] use copy mechanism [55], [56] to incorporate recommendation results into the generated responses. However, as it needs additional networks to predict when to switch between generation mode and copy mode, it may sometimes perform wrongly due to insufficient data to learn the pattern.

Inspired by [2], we use a simpler but effective integration mechanism. We directly add a vocabulary bias to the top decoder predictions. Different from their work, our vocabulary bias directly comes from the recommendation probabilities \( p_{rec} \) in (10):

\[
b_u = [0; \mathcal{G}(p_{rec})]
\]

where \( \mathbf{0} \) is a \( \mathcal{V} \)-dimensional zero vector, \( \mathcal{G}(\cdot) \) is an index selection operation to select items from entities \( \mathcal{G}(\cdot): \mathbb{R}^{|\mathcal{I}|} \rightarrow \mathbb{R}^{|\mathcal{I}|} \), and \([; ;]\) means concatenation. This makes the bias has the same dimension as our generation output (i.e., \(|\mathcal{V}| = |\mathcal{V}| + |\mathcal{I}| \)).

We dynamically add the bias \( b_u \) during generation based on the top predictions in each time stamp \( t \). Only when the top predictions of that time stamp include items in item set \( \mathcal{I} \) will we add the bias to them. The effect of this operation is that for a generation token \( y_t \) that is probably an item in \( \mathcal{I} \), we add its recommendation probability \( p_{rec}(y_t) \) to the original generation probability \( p(y_t) \). In this way, the results of the two modules are integrated so that our system can generate the recommendation-aware responses.

The total objective for our model is:

\[
\mathcal{L} = \mathcal{L}_{\text{gen}} + \gamma \mathcal{L}_{\text{rec}}
\]

where \( \gamma \) is a hyper-parameter that balances the two objectives. We jointly train the two objectives so that the two modules can benefit each other.

### IV. Experimental Setup

In this section, we first show the basic statistics of the two used datasets and the evaluation metrics for recommendation and generation modules in Section IV-A. Then we describe the baseline models together with some variants to our main model in Section IV-B. Finally, we list the parameter settings of our model in Section IV-C.

#### A. Datasets and Evaluation

**Datasets:** To empirically evaluate the proposed approach, we conduct experiments on two popular CRS datasets, namely, ReDial [1] and OpenDialKG [5]. ReDial is centered around movie recommendation and is constructed on Amazon Mechanical Turk (AMT) platform with pair crowd-workers (i.e., Seeker and Recommender). OpenDialKG consists of conversations that are mainly in four domains: movie, book, sports and music. The construction process of OpenDialKG is described as follows, which can reveal why the recommendation on OpenDialKG is much easier than on ReDial: The first agent initiates a conversation by giving a seed entity. The second agent is given with a list of facts that are relevant to the seed entity, and asked to choose the most relevant facts and use them to frame a free-form conversational response. Table II presents basic statistics about the two datasets.

As can be seen, the both datasets have around 10K conversations. We can notice that the average conversation length of ReDial dataset is larger than that of OpenDialKG. For the experiments, we split the two datasets into training, validation and test at ratios of 80%:10%:10% and 75%:15%:15% by following [1] and [5], respectively.

**Evaluation Metrics:** We evaluate the performance of recommendation and generation separately. For recommendation, we adopt Recall@K scores where \( K = 1, 10, 50 \) for ReDial by following [2], and \( K = 1, 3, 5, 10, 25 \) for OpenDialKG by following [5]. Recall@K indicates whether the predicted top-K items contain the ground truth recommendation items. For generation, apart from Dist-n (n=2, 3, 4) and perplexity (PPL) scores reported in previous work, we also report the case-insensitive BLEU-n (n=2, 4) scores.\(^2\) BLEU-n score that concerns with the probability of a sentence being produced by the model can reflect the fluency and coherence of the generated sentences, and Dist-n score ensures that the generated responses are not only coherent but also diverse. For a fair comparison, we calculate the PPL scores via a widely used off-the-shelf package KenLM,\(^3\) as the PPL scores may be very different when using different vocabulary.

### B. Baselines and Variants of Our Model

For the ReDial dataset, we compare our model with six competitive baselines proposed by the previous work:

- **ReDial [1]:** The model consists of a dialog generation module which is based on HRED [57], a recommendation module that is based on an auto-encoder [58] and a sentiment analysis module. Switching Mechanism is adopted to combine the recommendation and generation results.

- **KBRD [2]:** It utilizes DBpedia knowledge graph to enhance the semantic information of items (or entities) to do the recommendation and adopts a transformer based generation module, where knowledge graph information serves as word bias to assist the generation.

- **CRWalker [59]:** The method walks on the knowledge graph to form a reasoning tree at each turn for recommendation, and then maps to dialog acts to guide response generation. The learned recommendation-oriented dialog policy on the knowledge graph enhances the mutual benefit between recommendation and conversation.

\(^2\) We use NLTK package (https://www.nltk.org) to calculate the BLEU scores.

\(^3\)[Online]. Available: https://kheafield.com/code/kenlm/
TABLE III
RECOMMENDATION RESULTS (IN %) ON REDIAL DATASET

| Models          | Context | TextEntity | DBPedia | ConceptNet | Review | Rec@1  | Rec@10 | Rec@50 |
|-----------------|---------|------------|---------|------------|--------|--------|--------|--------|
| Baselines       |         |            |         |            |        | 2.4†  | 14.0†  | 32.0†  |
| REDIAL          | ✔       | ✔          | ✔       |            |        | 3.1†  | 15.0†  | 33.6†  |
| KBRD            | ✔       | ✔          | ✔       |            |        | 3.1†  | 15.5†  | 36.5†  |
| CRWALKER        | ✔       | ✔          | ✔       |            |        | 3.9†  | 18.3†  | 37.8†  |
| KGSF            | ✔       | ✔          | ✔       | ✔          |        | 4.6†  | 22.0†  | 39.6†  |
| REVCore         | ✔       | ✔          | ✔       | ✔          |        | 5.3   | 23.3†  | 40.7   |
| C² – CRS        | ✔       | ✔          | ✔       |            | ✔      | 3.0   | 16.4   | 35.0   |
| BART(CONTEXTM)  | ✔       | ✔          | ✔       |            | ✔      | 4.6   | 18.3   | 34.1   |
| ENTITYM-TIMEA   | ✔       | ✔          | ✔       |            | ✔      | 5.2   | 18.2   | 34.6   |
| ENTITYM-TIMEA-CONTEXTM | ✔  | ✔         | ✔       |            | ✔      | 5.9   | 24.0   | 41.3   |
| ENTITYM-SELFIA  | ✔       | ✔          | ✔       |            | ✔      | 3.3   | 16.3   | 32.6   |
| ENTITYM-TIMEA   | ✔       | ✔          | ✔       |            | ✔      | 5.7   | 22.6   | 40.4   |
| ENTITYM-TIMEA-CONTEXTM | ✔  | ✔         | ✔       |            | ✔      | 5.2   | 18.2   | 34.6   |

Our full model achieves the best performance with less external domain-specific knowledge as input, compared to the previous work. Baseline results marked with † and * indicate that our full model (the last row) performs significantly better than them (paired t-test: †; p < 0.05; *; p < 0.01).

- **KGSF** [3]: It uses mutual information maximization (MIM) [60] to align the semantic spaces of word- and entity-level KGs for the recommendation module. Its generation module includes a transformer encoder followed by a fused knowledge-enhanced decoder.
- **RevCore** [4]: It performs review-enriched and entity-based recommendation and uses a review-attentive encoder-decoder for generation. We use the re-implementation results in [51] for a fair comparison.
- **C²-CRS** [51]: This method extracts and represents the associated multi-grained semantic units from different data signals, and then align the corresponding semantic units from different data signals in a coarse-to-fine way.
- **seq2seq** [61]: It applies a seq2seq approach for entity path generation, given all of the dialog contexts.
- **Tri-LSTM** [62]: It encodes each utterance and all of its related facts within 1-hop from a KG to retrieve a response from a small (N=10) pre-defined sentence bank.
- **Ext-ED** [63]: It generates response entity token at its final softmax layer, and does not utilize structural information from knowledge graph.
- **DialKG Walker** [5]: It has an attention-based graph decoder that penalizes decoding of unnatural paths which prunes candidate entities and paths from a large search space.
- **BART (CONTEXTM)**: It only uses contextual-level representation (described in Section III-C-2) to do the recommendation. No entity-level representations and extra knowledge graph based knowledge are employed.
- **ENTITYM-TIMEA**: ENTITYM-TIMEA refers to entity modeling with our designed time-aware modeling. It also only uses entity-level representations.
- **ENTITYM-TIMEA-CONTEXTM**: Our full model that adopts both entity-level and contextual-level representation.

Besides, to examine the effectiveness of adding text entities to input (defined in Section III-C-1) for recommendation, we also evaluate our model variants with and without text entities (i.e., TextEntity in Table III) as input.

### C. Parameter Setting

For the RGCN-based recommendation module, we set both the entity embedding size and the hidden representation dimension to 128. The layer number for R-GCN is 1 and the normalization factor $Z_{e,r}$ is set to 1 following [2]. For the BART-based generation module, we finetune on the pre-trained BART-Base model, which consists of 6 layers of encoder and decoder, respectively. The hidden dimension, feed-forward network size and attention head number are 768, 3072 and 12 for both encoder and decoder layers. All the parameters of the original BART model, R-GCN module and the MLP layer for contextual-level prediction are learnable, which results in the total parameter number of 269M. The time recency effect $\lambda$ in (7), the balance factor between entity- and contextual-level representations in (10) and the tradeoff between two training objectives in (13) are set to 1.5, 0.5 and 1.0, respectively. We adopt diverse beam search [64] mechanism in generation with a beam size of 4 and the diverse beam group number is set to 2. All the hyper-parameters are determined by grid-search based on the validation performance.

We implement our models based on FAIRSEQ framework [65], and train on an NVIDIA 3090 GPU. During training, we set the max tokens of each batch to 4096 with an update
frequency of 4. We adopt Adam optimizer with a 5e-3 learning rate (and 5e-5 learning rate for BART-related modules as they are pre-trained parameters) and 1000 warm-up updates followed by a polynomial decay scheduler. The training time of one epoch is around 22 minutes. The model needs around 5 epochs to achieve the best performance on the validation set.

V. EXPERIMENTAL RESULTS

In this section, we first report the main comparison results on recommendation and generation in Sections V-A and V-B, respectively. Then we analyze the effectiveness of our proposed modules in Section V-C followed by a case study in Section V-D. Finally, we give more analysis in Section V-E.

A. Recommendation Result Comparisons

In this subsection, we present the main comparison results on ReDial dataset in Section V-A-1. Then, to further verify the effectiveness of our model in multi-domain dataset, we conduct experiments on OpenDialKG dataset and report the comparison results in Section V-A-2.

1) Results on ReDial: Table III shows the recommendation results of the baselines and the variants of our model on ReDial Dataset. We can draw the following observations from the results:

- Previous work tends to add more external knowledge, which can improve the recommendation performance, but ignores fully extracting the internal information. We can see that among the baselines, KBRED and CRWALKER add the entity-level knowledge (EntityKG, i.e., DBPedia and the text entities extracted from it) as input. KGSR further introduces word-level knowledge (WordKG, i.e., ConceptNet), while REVCore and C^2—CRS further incorporate item reviews (Review). All the added external knowledge introduces considerable improvement, demonstrating the effectiveness of the external knowledge. However, we can also notice that our EntityYM-TIMEA (Entity Modeling with Time-aware Attention) model only with the input of “Context” and “DBPedia” can also get a 4.6% Recall@1 score which is comparable with the results of REVCore, which utilizes two more external knowledge (Text Entity and Review). This shows that the previous designed methods cannot fully extract the internal knowledge from context, but our proposed model can capture more useful information for recommendation.

- Time-aware attention can better summarize user preference than self-attention mechanism. Our models with time-aware attention perform significantly better than the model with self-attention. For example, ENTITYYM-TIMEA achieves 1.9% higher Recall@1 than ENTITYYM-SELEA with the same input (Context + TextEntity + DBPedia). This validates our intuition that the self-attention mechanism which lacks explicit guidance lead to sub-optimal recommendations and the recently appeared items are more important in reflecting user preference. It also shows the effectiveness of our designed time-aware attention. More analysis about it can be found in Section V-E-1.

- BART-based contextual-level representations are helpful. We are the first to finetune a pre-trained BART model and utilize its representations for recommendation. As we can see in Table III, the simplest BART(CONTEXTM) model with only context as input achieves 35.0% Recall@50 while the KBRD model that incorporates DBPedia and text entities gets 33.6% Recall@50. We can also find that our models with time-aware attention show good improvements in all metrics after being enhanced with BART representations (i.e., with CONTEXTM). Both indicate that contextual-level representations extracted by the finetuned BART models can reflect user preferences that cannot be captured by entity-level representations, and so complement them well. Fig. 2(a) in Section V-C-1 shows more detailed analysis.

- Text entities are effective in capturing most relative items. Our model ENTITYYM-TIMEA with additional text entities (i.e., TextEntity) as input can achieve better Recall@1 compared with the same model without text entities (5.2% vs 4.6%), while keeping similar Recall@10 and Recall@50. This means that text entities help re-rank the top predictions and find the most relative items for recommendation.

2) Results on OpenDialKG: Apart from the ReDial that focuses on movie recommendation, we also examine our recommendation performance in a multi-domain dataset, OpenDialKG, to show the generalizability of our model. The results are displayed in Table IV. Two observations can also be drawn as follows (We skip some of the same observations as Redial here).

- The results on OpenDialKG are much higher than those on ReDial. As can be seen, our model achieves 18% Recall@1 score on OpenDialKG, which is much higher than 5.9% on ReDial. The reasons can be twofold. First, OpenDialKG is constructed based on the knowledge graph by some walking strategies [5], which means that the item prediction would be easier when incorporating entity-level knowledge. Second, we can see that the conversations on OpenDialKG are much shorter than those on ReDial (see Table II). This indicates that it takes less turns leading to the final recommendation needed by the seekers and so the modeling of the appeared items on OpenDialKG may be simpler.
Our model generalizes well to multiple domains. As we can see from Table IV, our model with time-aware attention and BART-enhanced representations (i.e., ENTITYM-TIMEA-CONTEXTM) achieves the best performance compared with all the other methods. We can observe similar trends among the different variants as those on ReDial, e.g., time-aware attention is better than self-attention and BART representations help improve all of the metrics. This validates that our method generalizes well to other domains.

### B. Generation Result Comparisons

In this subsection, we first discuss the automatic evaluation results in Section V-B.1 and then adopt a human evaluation to examine the generation results from a different perspective in Section V-B.2.

1) **Automatic Evaluation:** We show the automatic evaluation comparison results on ReDial in Table V. To investigate the performance in different scenarios, we display the results of our models with conventional beam search (BS) and diverse beam search (DBS), respectively, together with their results without BART pre-training. Our model yields significantly better results on all of the evaluation metrics (including Dist-n, BLEU-n and perplexity) than the baselines, which indicates that our model tends to produce more diverse, coherent and fluent responses. In the following, we detail our observations:

- **Our model generalizes well to multiple domains.** As we can see from Table IV, our model with time-aware attention and BART-enhanced representations (i.e., ENTITYM-TIMEA-CONTEXTM) achieves the best performance compared with all the other methods. We can observe similar trends among the different variants as those on ReDial, e.g., time-aware attention is better than self-attention and BART representations help improve all of the metrics. This validates that our method generalizes well to other domains.

- **Our model is able to generate more diverse, coherent and fluent responses than the baselines.** As can be seen, our model with beam search achieves the best BLEU scores and perplexity, and our model with diverse beam search yields the highest Dist-n while maintaining comparable BLEU scores and perplexity. Comparing our models with and without BART pre-training, the Dist-n metrics are affected severely, which validates the effectiveness of BART pre-training in enhancing the diversity of generated responses.

- **It is challenging to balance between diversity and fluency.** The baselines tend to perform differently in terms of Dist-n and perplexity, e.g., KGSF achieves higher Dist-n than KBRD, but its perplexity is worse. We presume that higher diversity requires the models to extract more diverse patterns to express their content, but organizing them into a fluent response may be challenging. Another example is our model without BART pre-training achieves poor diversity (i.e., very low Dist-n scores), as it may overfit in the training corpus and tend to generate simple responses. But this also results in its lowest perplexity. Our model with diverse beam search achieves consistently better Dist-n scores, and maintains relatively lower perplexity, demonstrating the superiority of our model design.

- **The Dist-n scores highly depend on the searching strategy.** Our model performs much better in terms of Dist-n when applying diverse beam search rather than conventional beam search. What is more, different configurations (e.g., length penalty in Section V-B.2) applied during response generation also affects the scores. This shows that it is not enough to evaluate the generation performance of these models only based on the metric, so we also include BLEU scores here and human evaluation (Section V-B.2) for better evaluation.

2) **Human Evaluation:** To conduct our human evaluation, we randomly sampled 100 context-response pairs from the test set and collected the corresponding generation results of our models as well as the baselines. We then employ two crowd-workers to score the results on the scale of [0, 1, 2], where higher scores indicate better quality. Following [66], we evaluate the following three aspects of the results:

- **Fluency:** whether a response is in a proper English grammar and easy to understand.

- **Informativeness:** whether a response contains meaningful information. The “safe responses” are treated as uninformative as they may be repetitive and meaningless.

- **Coherence:** whether a response is coherent with its previous context, i.e., the discussed content should be consistent.

We display the human evaluation results in Table VI. As can be seen, all the models generate responses with high fluency, but perform differently regarding informativeness and coherence. The baselines are more likely to produce safe responses (short and repetitive), resulting in lower informativeness. Also, their coherence scores are lower since they might produce some meaningless responses no matter what the previous contexts are. Nevertheless, our model can generate more informative and coherent responses.

### C. Effectiveness of the Proposed Mechanisms

In this subsection, we first test the recommendation results with varying mentioned items in Section V-C.1 to show the effectiveness of the introduced contextual-level representation. Then we discuss the effects of BART pre-training for our model in Section V-C.2.

1) **Effectiveness of Contextual-Level Representation:** To investigate how contextual-level representations influence recommendation, we show the Recall@50 scores on ReDial with varying mentioned items numbers in context in Fig. 2(a) for our three model variants, CONTEXTM, ENTITYM-TIMEA and ENTITYM-TIMEA-CONTEXTM. The same experiment is also conducted on OpenDialKG, where results are displayed in Fig. 3(a) and Recall@25 scores are reported. The mentioned item number means how many items (e.g., movie, books) appear in the history utterances. Specifically, we divide the test samples into several groups based on the mentioned item number and calculate recall score for each group.
TABLE V

**Generation Results (in %) on the Redial Dataset**

| Models        | Dist-2 | Dist-3 | Dist-4 | BLEU-2 | BLEU-4 | Perplexity↓ |
|---------------|--------|--------|--------|--------|--------|------------|
| Transformer   | 14.8   | 15.1   | 13.7   | -      | -      | -          |
| ReDial        | 22.5   | 23.6   | 22.8   | 17.8   | 7.4    | 61.7       |
| KBRD          | 26.3   | 36.8   | 42.3   | 18.5   | 7.4    | 58.8       |
| KGSF          | 28.9   | 43.4   | 51.9   | 16.4   | 7.4    | 131.1      |
| RevCore       | 42.4   | 55.8   | 61.2   | -      | -      | -          |
| Ours+BS       | 35.8   | 49.9   | 57.7   | 19.1   | 9.3    | 52.1       |
| - w/o BART Pre-train | 8.5   | 10.9   | 12.3   | 18.6   | 8.7    | 30.7       |
| Ours+DBS      | 45.7   | 68.3   | 76.1   | 19.1   | 8.9    | 54.8       |
| - w/o BART Pre-train | 13.9   | 19.8   | 23.8   | 18.6   | 8.2    | 43.9       |

"BS" refers to beam search, “DBS” refers to diverse beam search.

TABLE VI

**Human Evaluation of the Generation Results on the Redial Dataset**

| Models       | Fluency | Informativeness | Coherence |
|--------------|---------|-----------------|-----------|
| HUMAN        | 1.95    | 1.71            | 1.71      |
| ReDial       | 1.92    | 1.32            | 1.23      |
| KBRD         | 1.95    | 1.39            | 1.31      |
| KGSF         | 1.91    | 1.02            | 0.95      |
| Ours+BS      | 1.95    | 1.54            | 1.66      |
| Ours+DBS     | 1.92    | 1.64            | 1.64      |

"BS" refers to beam search, “DBS” refers to diverse beam search. All the metrics are in the scale of [0, 2]. The overall Cohen’s kappa coefficient is larger than 0.65.

We observe different trends for the two datasets. From Fig. 2(a), CONTEXTM model performs better than ENTITYM-TIMEA when the mentioned number is 0 or 1. Such a phenomenon is desired since when the item history information is rare or even missing, the entity-level representations are not sufficient to produce reliable recommendations while the contextual-level representations can capture useful information from the text. However, we can find that the trend in OpenDialKG is different as indicated in Fig. 3(a). All models perform extremely worse when the mentioned number is 0. This might be due to the different data distributions of the two datasets, where little useful information is presented at the very beginning of the conversations in OpenDialKG.

Then we find ENTITYM-TIMEA performs consistently better than CONTEXTM when the mentioned number increases. Because the increasing mentioned item number also means the context becomes longer, and the model might not be able to handle the long context, especially when the mentioned item number is larger than 10. This means that contextual-level representations are useful in the cold-start scenario (not a rare situation in Redial as shown in Fig. 2(b)), which is a shortcoming for entity-level recommendation. Therefore, our full model (i.e., ENTITYM-TIMEA-CONTEXTM) combines the both representations and yields the best performance in all the situations for both datasets.

2) Effects of BART Pre-Training: We elaborate on the effects of BART pre-training (PT) here. For recommendation, we list the recall scores of our different model variants with and without BART PT in both Tables VII (Redial) and VIII (OpenDialKG).

As can be seen, most of the models perform worse when removing pre-training, as pre-training on large-scale monolingual datasets helps the models learn better semantic features. The exception is the CONTEXTM model in OpenDialKG, where PT contributes most on Recall@1 but has fewer effects on the rest metrics. A possible reason is that the conversation patterns in OpenDialKG might be very different from the pre-training corpus of BART model, resulting in less potency.

For generation, we have listed the ablation without pre-training in Table V. As we can see, models without pre-training show poor performance on the Dist-n metrics as models tend to be overfitting and cannot generate diverse responses based on a small dataset.

D. Case Study

To give an intuitive assessment of the quality of the generated sentences and recommended items, we present a conversation example in Table IX. Apart from the results generated by our model and the human ground-truths from the corpus, we list the results of ReDial, KBRD and KGSF models. The reason why we do not list the results of RevCore and C2-CRS is their released codes are not complete to re-implement the generation results.

From Table IX, we can easily find that our model tends to generate responses with more diversity and coherence to the
TABLE VII
RECOMMENDATION RESULTS (IN %) OF OUR MODELS WHEN USING BART PRE-TRAINING (PT) OR NOT ON RE.DIAL DATASET

| Models                  | Inputs                  | Pre-training | R@1      | R@3      | R@5 | R@10     | R@50     |
|-------------------------|-------------------------|--------------|----------|----------|-----|----------|----------|
| BART (CONTEXTM)         | Context                 | ✓            | 3.0 (15.4%) | 16.4 (13.1%) | 35.0 (10.4%) | 2.6 | 14.5 | 31.7 |
| ENTITYM-TIMEA-CONTEXTM  | Context+DBpedia         | ✓            | 5.7 (9.6%) | 22.6 (5.6%) | 40.4 (1.3%) | 5.2 | 21.4 | 39.9 |
| ENTITYM-TIMEA-CONTEXTM  | Context+TextEntity+DBpedia | ✓       | 5.9 (7.3%) | 24.0 (9.1%) | 41.3 (2.5%) | 5.5 | 22.0 | 40.3 |

The values in parentheses correspond to the percentage increase after adding PT.

TABLE VIII
RECOMMENDATION RESULTS (IN %) OF OUR MODELS WHEN USING BART PRE-TRAINING (PT) OR NOT ON OPEN.DIAL.KG DATASET

| Models                  | Inputs                  | Pre-training | R@1      | R@3      | R@5  | R@10     | R@25     |
|-------------------------|-------------------------|--------------|----------|----------|------|----------|----------|
| BART (CONTEXTM)         | Context                 | ✓            | 5.8 (5.5%) | 19.7 (1.0%) | 31.0 (-1.9%) | 45.5 (-0.9%) | 57.8 (-0.3%) | 5.5 | 19.5 | 31.6 | 45.9 | 58.0 |
| ENTITYM-TIMEA-CONTEXTM  | Context+DBpedia         | ✓            | 18.0 (4.0%) | 33.5 (6.0%) | 41.5 (7.8%) | 50.0 (2.2%) | 64.8 (1.2%) | 17.3 | 31.6 | 38.5 | 48.9 | 64.0 |

The values in parentheses correspond to the percentage increase after adding PT.

TABLE IX
A CONVERSATION EXAMPLE ON MOVIE ITEM RECOMMENDATION

Seeker: I have seen The Conjuring (2013) I very much enjoyed this movie. When I was younger I liked movies like Friday the 13th (1980).

Human: I liked that one too!
ReDial: I have n’t seen that one. I’ve heard good things about it.
KBRD: I have n’t seen that one. I’ve heard Halloween (1978) was good.
KGSF: How about The Deacont (2005) ?
OUR: I haven’t seen that one, but I’ve heard good things about it.

Seeker: I also liked to watch the A Nightmare on Elm Street (1984) Series.

Human: I loved that as well mainly because Johnny Depp is in it
ReDial: I have n’t seen that one yet. I’ve seen It (2017) .
KBRD: I have n’t seen that one. I’ve heard Halloween (1978) was good.
KGSF: I also recommend Insideus (2011) .
OUR: I liked that one too! I also recommend The Last Exorcism (2010).

Seeker: I also liked Johnny Depp? So handsome, such a great actor.
I love Johnny Depp in Alice in Wonderland (2010).

Human: Me too one of my favorites from him is The Astromaut’s Wife (1999)
ReDial: I have n’t seen that one yet. I’ve seen It (2017) .
KBRD: I have n’t seen that one. I will check it out.
KGSF: I hope you enjoy them!
OUR: Yes, I love Johnny Depp too! I have seen the new It (2017) movie?

The responses of ReDial, KBRD and KGSF are from the test results of the corresponding models. Human responses are the ground-truths from the corpus.

In this subsection, we extensively explore the effects of some hyper-parameters, together with in-depth discussions.

1) Influence of Recency Effect in Time-Aware Model: We present the Recall@1 and Recall@10 (Recall@50 shows a similar trend with Recall@10 so we omit here) scores with varying values of $\lambda$ in Fig. 4(a), which controls the effect of recency introduced in our proposed Time-Aware Attention (see (7)). $\lambda < 1$ indicates the earlier appeared items are more important, which results in quick performance drops. And the performance increases when $\lambda > 1$ compared to $\lambda = 1$. This validates the intuition that more recently appeared items contribute more to the next item recommendation. On the other hand, Recall@1 and Recall@10 present different trends when $\lambda$ increases – Recall@1 consistently increases while Recall@10 begins to drop when $\lambda > 2$. This is because when $\lambda$ is too large, time-aware attention tends to concern with the most recently appeared item. This is helpful in finding the most relative items but hurts the overall recommendation.

2) Limitation of Dist-N Metrics: As we find that search strategies seriously affect the Dist-n metrics (see Section V-B-1), we present more analysis on them by exploring the length penalty (a hyper-parameter that can control the lengths of final generated results) when generating the responses. We display the results of Dist-2 and BLEU-2 for our ENTITYM-TIMEA-CONTEXTM model with different length penalty in Fig. 4(b) (other metrics are in similar trends). We can find that the generated lengths also affect a lot the Dist-n scores since longer responses allow more different tokens to be generated.
Fig. 4. Effects of hyper-parameters on ReDial: parameter $\lambda$ to control the time-aware user preference modeling, length penalty to influence the generation, and parameter $\mu$ to balance the effects of entity- and contextual-level user representations.

Fig. 5. Recall@50 of our full model with varying hyper-parameter $\mu$ over the number of mentioned items on the ReDial dataset.

However, this is not expected as the longer is not necessarily the better. Therefore, other metrics, including human evaluation, are desired to explicitly evaluate generation performance.

3) Trade-Off Between Entity-Level and Contextual-Level Representations: We examine the effects of the trade-off hyper-parameter $\mu$ in (10) by setting its value from 0 (only entity-level representations) to 1 (only contextual-level representations) and display the results of our ENTITYM-TIMEA-CONTEXTM model in Fig. 4(c). As can be seen, Recall@50 score is significantly improved when the value of $\mu$ changes from 0 to 0.1 (or 1 to 0.9). This validates that the two representations can capture user preferences from different perspectives and complement each other for better recommendation performance. Besides, the trends of Recall@1 and Recall@10 are similar to that of Recall@50. The best results are achieved around $\mu = 0.5$, showing that both levels of representation are important.

We further demonstrate the performance of our full model using varying $\mu$ values (0.0, 0.2, 0.5, 0.8, 1.0) across different numbers of mentioned items in the ReDial dataset, and present the results in Fig. 5. Considering that $\mu = 0.0$ implies recommendations solely based on context (i.e., BART-based contextual modeling), we can observe that the performance of model with $\mu = 0.0$ degrades as number of item mentions grows. And the models with $\mu > 0.0$ show a steady trend.

4) Comparing Different Models for Context Representation: As observed in (9), we compute the average of context representations across sequences to derive contextual-level representations for context $C$, identified as AVG method. In this subsection, we explore two different context modeling methods, including LSTM, and ATTN. For LSTM, we add an additional LSTM layer on top of the context representations $H^C$ and use the representation of the last hidden state as the final contextual-level representation; for ATTN, we use a learnable query vector to learn the attention weights for each context representation in $H^C$, and then their weighted average serves as the final contextual-level representation. We present the recommendation results alongside parameter sizes and inference times in Table X. From Table X, it is evident that our simple averaged context representation, noted for having the fewest parameters and the fastest inference time, can yield comparable recommendation performance.

VI. CONCLUSION AND FUTURE WORK

In this work, we focus on conversational recommender systems and propose to capture both entity-level and contextual-level representations to improve the recommendation performance, where a time-aware user preference modeling is designed to better capture the user’s interests and a pre-trained BART model is used to enhance contextual-level user preference modeling and improve the diversity of the generated responses. Experiments on two publicly available datasets show that the proposed model can achieve better performance with less external domain-specific knowledge and generalizes well to other domains. Further analyses also examine the effectiveness of our model in different scenarios.
In future work, we would like to explore more accurate user preference modeling mechanisms. Though our current time-aware modeling method shows promising improvement on both datasets, it is based on a simple assumption and cannot deal with more complicated real-world scenarios. A combined mechanism that concerns time effects and other features like sentiment is desired. However, it is hard to train a powerful user preference learning framework with the currently available datasets, which are human-created and relatively small. Additionally, we want to highlight that our investigation into conversation recommendations currently overlooks the user’s historical purchasing and rating patterns. This oversight, coupled with the absence of user-level interests, results in recommendations primarily reliant on the ongoing conversation context, without due consideration for individual user profiles and past behaviors. Thus, it might be imperative to acquire a larger, more diverse dataset encompassing user profiles, historical data, and spanning multiple domains in order to effectively address these limitations.

ACKNOWLEDGMENTS

We express our gratitude to the editors and all the anonymous reviewers for their insightful comments and suggestions.

REFERENCES

[1] R. Li, S. Kahou, H. Schulz, V. Michalski, L. Charlin, and C. Pal, “Towards deep conversational recommendations,” 2018, arXiv:1812.07617.
[2] Q. Chen et al., “Towards knowledge-based recommender dialog system,” in Proc. Conf. Empirical Methods Natural Lang. Proc. 9th Int. Joint Conf. Natural Lang. Process. (Hong Kong, China, 2019, pp. 1803–1813). [Online]. Available: https://aclanthology.org/D19-1189
[3] K. Zhou, W. X. Zhao, S. Bian, Y. Zhou, J.-R. Wen, and J. Yu, “Improving conversational recommender systems via knowledge graph based semantic fusion,” in Proc. 26th ACM SIGKDD, 2020, pp. 1006–1014.
[4] Y. Lu et al., “Recover: Review-augmented conversational recommendation,” 2021, arXiv:2106.00957.
[5] S. Moon, P. Shah, A. Kumar, and R. Subba, “OpenDialogKG: Explorable conversational reasoning with attention-based walks over knowledge graphs,” in Proc. 57th ACL, 2019, pp. 845–854.
[6] M. Lewis et al., “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, 2020, pp. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703
[7] L. Wang, H. Hu, L. Sha, C. Xu, K.-F. Wong, and D. Jiang, “Finetuning large-scale pre-trained language models for conversational recommendation with knowledge graph,” 2021, arXiv:2110.07477.
[8] G. Adomavicius and A. Tuzhilin, “Context-aware recommender systems,” in Recommender Systems Handbook. Berlin, Germany: Springer, 2010, pp. 217–253.
[9] S. Raza and C. Ding, “Progress in context-aware recommender systems—An overview,” in Comput. Sci. Rev., vol. 31, pp. 84–97, 2019.
[10] S. Raza and C. Ding, “A regularized model to trade-off between accuracy and diversity in a news recommender system,” in Proc. IEEE Int. Conf. Big Data, 2020, pp. 551–560.
[11] K. Christakopoulou, A. Beutel, R. Li, S. Jain, and E. H. Chi, “Q&R: A two-stage approach toward interactive recommendation,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2018, pp. 139–148.
[12] Y. Zhong, X. Chen, Q. Ai, L. Yang, and W. B. Croft, “Towards conversational search and recommendation: System ask, user respond,” in Proc. 27th ACM Int. Conf. Inf. Inf. Manage., 2018, pp. 177–186.
[13] M. Aliannejadi, H. Zamani, F. Crestani, and W. B. Croft, “Asking clarifying questions in open-domain information-seeking conversations,” in Proc. 42nd Int. ACM SIGIR Conf., 2019, pp. 475–484.
[14] W. Liu et al., “Estimation-action-reflection: Towards deep interaction between conversational and recommender systems,” in Proc. 13th Int. Conf. Web Search Data Mining, 2020, pp. 304–312.

[15] X. Ren et al., “CRSAL: Conversational recommender systems with adversarial learning,” ACM Trans. Inf. Syst., vol. 38, no. 4, pp. 1–40, 2020.
[16] Y. Deng, Y. Li, F. Sun, B. Ding, and W. Lam, “Unified conversational recommendation policy learning via graph-based reinforcement learning,” 2021, arXiv:2105.00710.
[17] H. Xu et al., “User memory reasoning for conversational recommendation,” in Proc. 28th Int. Conf. Comput. Linguistics, Barcelona, Spain, 2020, pp. 5288–5308. [Online]. Available: https://aclanthology.org/2020.coling-main-463
[18] W. Lei et al., “Interactive path reasoning on graph for conversational recommendation,” in Proc. 26th ACM SIGKDD Conf. Knowl. Discov. Data Mining, Virtual Event, CA, USA, 2020, pp. 2073–2083.
[19] X. Ren, H. Yin, T. Chen, H. Wang, Z. Huang, and K. Zheng, “Learning to ask appropriate questions in conversational recommendation,” 2021, arXiv:2105.04774.
[20] K. Xu, J. Yang, J. Xu, S. Gao, J. Guo, and J.-R. Wen, “Adapting user preference to online feedback in multi-round conversational recommendation,” in Proc. 14th ACM Int. Conf. Web Search Data Mining, 2021, pp. 364–372.
[21] L. Li, W. Chu, J. Langford, and R. E. Schapire, “A contextual-bandit approach to personalized news article recommendation,” in Proc. 19th Int. Conf. World Wide Web, 2010, pp. 661–670.
[22] S. Li, A. Karatzoglou, and C. Gentile, “Collaborative filtering bandits,” in Proc. 39th Int. ACM SIGIR, 2016, pp. 539–548.
[23] K. Christakopoulou, F. Radinsky, and K. Hofmann, “Towards conversational recommender systems,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 815–824.
[24] S. Li, W. Lei, Q. Wu, X. He, P. Jiang, and T.-S. Chua, “Seamlessly unifying attributes and items: Conversational recommendation for cold-start users,” 2020, arXiv:2005.12971.
[25] Y. Sun and Y. Zhang, “Conversational recommender system,” in Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2018, pp. 235–244.
[26] K. Zhou, Y. Zhou, W. X. Zhao, X. Wang, and J.-R. Wen, “Towards topic-guided conversational recommender system,” 2020, arXiv:2010.04125.
[27] D. Kang, A. Balakrishnan, P. Shah, P. Crook, Y.-L. Bourreau, and I. Weston, “Recommendation as a communication game: Self-supervised bot-play for goal-oriented dialogue,” 2019, arXiv:1909.03922.
[28] Z. Liu, H. Wang, Z.-Y. Niu, H. Wu, W. Che, and T. Liu, “Towards conversational recommendation over multi-type dialogs,” 2020, arXiv:2005.09354.
[29] S. A. Hayati, D. Kang, Q. Zhu, W. Shi, and Z. Yu, “Inspired: Towards scalable recommendation dialog systems,” 2020, arXiv:2009.14306.
[30] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, “A review on deep learning for recommender systems: Challenges and remedies,” Artif. Intell. Rev., vol. 52, no. 1, pp. 1–37, 2019.
[31] Y. Hu, Y. Koren, and C. Volinsky, “Collaborative filtering with implicit feedback datasets,” in Proc. IEEE 8th Int. Conf. Data Mining, 2008, pp. 263–272.
[32] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative filtering recommender systems,” in The Adaptive Web. Berlin, Germany: Springer, 2007, pp. 291–324.
[33] R. VanMeteren and M. VanSonmeren, “Using content-based filtering for recommendation,” in A. Mach. Learn. New Inf. Age: MLweek/ECMLEworkshop, 2000, pp. 47–56.
[34] T. Tran and R. Cohen, “Hybrid recommender systems for electronic commerce,” in Proc. Knowl.-Based Electron. Markets, Papers AAAI Workshop, 2000, pp. 78–84.
[35] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in Proc. 10th Int. Conf. World Wide Web, 2001, pp. 285–295.
[36] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” 2013, arXiv:1301.7363.
[37] L. H. Ungar and D. P. Foster, “Clustering methods for collaborative filtering,” in Proc. AAAI Workshop Recommendation Syst., Menlo Park, CA, USA, 1998, pp. 114–129.
[38] X. Su and T. M. Khoshgoftaar, “Collaborative filtering for multi-class data using belief nets algorithms,” in Proc. IEEE 18th Int. Conf. Tools Intell. Info., 2006, pp. 497–504.
[39] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer, vol. 42, no. 8, pp. 30–37, 2009.
[40] P. Lops, M. D. Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in Recommender Systems Handbook. Berlin, Germany: Springer, 2011, pp. 73–105.
Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003.

X. Wang and A. Gupta, “Unsupervised learning of visual representations using videos,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2794–2802.

F. Petroni et al., “Language models as knowledge bases?,” in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, Hong Kong, China, 2019, pp. 2463–2473. [Online]. Available: https://aclanthology.org/D19-1250

T. Gao, A. Fisch, and D. Chen, “Making pre-trained language models better few-shot learners,” 2020, arXiv:2012.15723.

X. Liu et al., “GPT understands, too,” 2021, arXiv:2103.10385.

Y. Zhou, K. Zhou, W. X. Zhao, C. Wang, P. Jiang, and H. Hu, “C2- CRS: Coarse-to-fine contrastive learning for conversational recommender system,” 2022, arXiv:2201.02732.

A. Vaswani, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst. 30: Annu. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 5998–6008.

J. Dauber, M. Jakob, C. Hokamp, and P. N. Mendes, “Improving efficiency and accuracy in multilingual entity extraction,” in *Proc. 9th Int. Conf. Semantic Web Conf.*, 2013, pp. 121–124.

M. Schlichtkrull et al., “Modeling relational data with graph convolutional networks,” in *Proc. Eur. Semantic Web Conf.*, 2018, pp. 593–607.

J. Gu, Z. Lu, H. Li, and V. O. Li, “Incorporating copying mechanism in sequence-to-sequence learning,” in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, 2016, pp. 1631–1640. [Online]. Available: https://aclanthology.org/P16-1154

C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, “Pointing the unknown words,” in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, 2016, pp. 140–149. [Online]. Available: https://aclanthology.org/P16-1014

A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, and J.-Y. Nie, “A hierarchical recurrent encoder-decoder for generative context-aware query suggestion,” in *Proc. 24th ACM Int. Conf. Knowl. Manag.*, 2015, pp. 553–562.

J. He, H. H. Zhuo, and J. Law, “Distributed-representation based hybrid recommender system with short item descriptions,” 2017, arXiv:1703.04854.

W. Ma, R. Takanobu, M. Tu, and M. Huang, “Bridging the gap between conversational reasoning and interactive recommendation,” 2020, arXiv:2010.10333.

P. Viola and W. M. WellsIII, “Alignment by maximization of mutual information,” *Int. J. Comput. Vis.*, vol. 24, no. 2, pp. 137–154, 1997.

I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence and K. Q. Weinberger, Eds., Curran Associates, Inc., vol. 27, 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/a4ca55a4f2747245cd8d9e4cc13c743d2-Paper.pdf

T. Young, E. Cambria, I. Chaturvedi, M. Huang, H. Zhou, and S. Biswas, “Augmenting end-to-end dialog systems with commonsense knowledge,” 2017, arXiv:1709.05453.

P. Parthasarathi and J. Pineau, “Extending neural generative conversational model using external knowledge sources,” 2018, arXiv:1809.05254.

A. K. Vijayarajkumar et al., “Diverse beam search: Decoding diverse solutions from neural sequence models,” 2016, arXiv:1610.02424.

M. Ott et al., “fairseq: A fast, extensible toolkit for sequence modeling,” 2019, arXiv:1904.01038.

S. Bao et al., “PLATO-2: Towards building an open-domain chatbot via curriculum learning,” 2020, arXiv:2006.16779.