5-2022

Topic-guided conversational recommender in multiple domains

Lizi LIAO
Singapore Management University, lzliao@smu.edu.sg

Ryuichi TAKANOBU

Yunshan MA

Xun YANG

Minlie HUANG

See next page for additional authors

Citation

LIAO, Lizi; TAKANOBU, Ryuichi; MA, Yunshan; YANG, Xun; HUANG, Minlie; and CHUA, Tat-Seng. Topic-guided conversational recommender in multiple domains. (2022). *IEEE Transactions on Knowledge and Data Engineering*. 34, (5), 2485-2496.
Available at: https://ink.library.smu.edu.sg/sis_research/7650

Topic-Guided Conversational Recommender in Multiple Domains

Lizi Liao, Ryuichi Takanobu, Yunshan Ma, Xun Yang, Minlie Huang, and Tat-Seng Chua

Abstract—Conversational systems have recently attracted significant attention. Both the research community and industry believe that it will exert huge impact on human-computer interaction, and specifically, the IR/RecSys community has begun to explore Conversational Recommendation. In real-life scenarios, such systems are often urgently needed in helping users accomplishing different tasks under various situations. However, existing works still face several shortcomings: (1) Most efforts are largely confined in single task setting. They fall short of hands in handling tasks across domains. (2) Aside from soliciting user preference from dialogue history, a conversational recommender naturally has access to the back-end data structure which should be fully leveraged to yield good recommendations. In this paper, we thus present a Topic-guided Conversational Recommender (*TCR*) which is specifically designed for the multi-domain setting. It augments the sequence-to-sequence (seq2seq) models with a neural latent topic component to better guide the response generation. To better leverage the dialogue history and the back-end data structure, we adopt a graph convolutional network (GCN) to model the relationships between different recommendation candidates while also capture the match between candidates and the dialogue history. We then seamlessly combine these two parts with the idea of pointer networks. We perform extensive evaluation on a large-scale task-oriented multi-domain dialogue dataset and the results show that our method achieves superior performance as compared to a wide range of baselines.

Index Terms—Conversational recommendation, topic modeling, graph convolutional networks

1 Introduction

Conversational systems such as Google Now, Apple Siri, and Microsoft Cortana serve as the direct interactive portal for end-users. It is expected to revolutionize the way of human machine interaction. Specifically, due to users' constant need to look for help to sail through huge amount of candidates and make choices, both the industry and research community such as IT/RecSys have swarmed into exploring conversational recommendation [1], [2], [3].

Although conversational recommenders show big potential, it is non-trivial to build such an intelligent system to meet the various user needs. First of all, users often expect such intelligent systems to be able to handle different tasks in various situations, where existing works fall short of hands. As the example illustrated in Fig. 1, it naturally involves several sub-tasks such as hotel reservation, restaurant booking and attraction recommendation *etc*. Thus the agent should have the ability to recognize those topics from the context and generate within-topic responses. However, current state-of-the-art methods might not be sufficient to achieve this. In general, neural conversational models [4], [5], [6] are the latest development in conversational modeling, where seq2seq-based models, such as HRED [6], are employed for generating responses in an end-to-end fashion. Such models are good at capturing the local structure of word sequence but might face difficulty in remembering global semantic structure of dialogue sessions.

Second, as shown in in Fig. 1, to satisfy users' need such as finding hotel or restaurant, the ability to make appropriate venue[1] recommendation considering various relationships is essential. In early efforts, task-oriented dialogue systems tried to form database queries based on user utterances to retrieve venues [7], [8], [9], [10], [11]. However, such methods heavily rely on the exact match of queries in database which is rather sensitive to even slight language variations. More importantly, the various relationships among venues and attributes are hard to be fully described via textual queries. Therefore, the most recent studies such as [12], [13] integrated conversational system with recommendation components which are more flexible and are able to capture more complicated interactions. Nonetheless, the recommendation part of these work only focus on learning the interplay between users and items (same as venues in this work). The interplay between venues and attributes as well as various relationships among venues fall out of consideration. However, such information is important for accurate venue recommendation especially in the multi-domain setting. For example, in order to generate the hotel 'cambridge belfry' as in Fig. 1, the agent needs to capture not only the useful information like '4 stars' and 'free wifi' via textual utterance, but also the hidden location constraint via venue 'TR3211'.

In this paper, we thus propose a Topic-guided Conversational Recommender (*TCR*) as shown in Fig. 2 and apply it to

- *Lizi Liao, Yunshan Ma, Xun Yang, and Tat-Seng Chua are with the National University of Singapore, Singapore 119077. E-mail: {liaolizi.llz, mysbupt, hfutyangxun}@gmail.com, chuats@comp.nus.edu.sg.*
- *Ryuichi Takanobu and Minlie Huang are with the Tsinghua University, Beijing 100084, China. E-mail: truthless11@gmail.com, aihuang@tsinghua.edu.cn.*

Manuscript received 14 June 2019; revised 21 June 2020; accepted 25 June 2020. Date of publication 10 July 2020; date of current version 1 Apr. 2022. (Corresponding author: Lizi Liao.)
Recommended for acceptance by E. Chen.
Digital Object Identifier no. 10.1109/TKDE.2020.3008563

1. The recommendation candidates are like venue names or train numbers in dataset. For ease of illustration, we all describe as venues throughout the paper.

U: Hi, I'm looking for a train that is going to cambridge and arriving there by 20:45, is there anything like that?
A: There are over 1,000 trains like that. Where will you be departing from?
U: I am departing from birmingham new street.
A: Can you confirm your desired travel day?
U: I would like to leave on Wednesday.
A: TR3211 is leaving birmingham new street at 17:40 and arriving at 20:23 on Wednesday. Will this work for you?
U: That will, yes. Please make a booking for 5 people please.
A: I've booked your train tickets, and your reference number is A9NHSO9Y.
Train
U: Thanks so much. I would also need a place to stay. I am looking for something with 4 stars and has free wifi.
A: How about the cambridge belfry? It has all the attributes you requested and a great name! Maybe even a real belfry?
U: That sounds great, could you make a booking for me please?
A: What day would you like your booking for?
U: Please book it for Wednesday for 5 people and 5 nights, please.
A: Booking was successful. Reference number is : 5NAWGJDC.
Hotel

Fig. 1. A sample dialogue between a user (**U**) and an agent (**A**) from the dataset. We observe the need for global topic control and accurate recommendation.

the multi-domain setting. First, in order to enable the agent to swiftly differentiate topics in various domains, we leverage the underlying seq2seq-based model to capture the local dynamics of utterances while extract and represent its global semantics by a mixture of topic components like topic models [14]. Second, we employ a graph convolutional network (GCN) to capture the various relationships between recommendation candidates (we denote as venues as the example in Fig. 1 thereafter) and attributes. It automatically learns the match between venue and dialogue context. When generating venue recommendations, the agent ranks the venues by calculating the matching scores between the learned venue representations and dialogue context representations. The key idea is that GCN-based component helps the conversational recommender to generate better representations of venues that incorporate both venue attribute information as well as venue relations. Third, we combine the topic-based component and the GCN part via leveraging the idea of pointer networks. It allows us to effectively incorporate the recommendation results into the response generation procedure. Extensive experiments are carried out on a large scale multi-domain task-oriented conversational dataset. The proposed method manages to achieve superior performance across baselines.

To sum up, the main contributions are threefold:

- We propose a conversational recommender which handles multiple sub-tasks involving seven topics — attraction, hospital, police, hotel, restaurant, taxi and train. A neural topic component helps it to generate within-topic responses by narrowing down the generation of tokens in decoding.
- We employ a GCN-based venue recommender which captures the interplay between venues and attributes as well as various relationships among venues. It also helps to learn the match between venues and the dialogue contexts. Inspired by pointer networks, an integration mechanism is used to incorporate the recommendation results to the final responses.
- We conduct extensive experiments to evaluate the proposed method under various evaluation metrics and show superior performance over the state-of-the-art methods.

In the rest of the paper, we review related work in Section 2. Section 3 describes the elementary building blocks of the proposed learning method. Experimental results and analysis are reported in Section 4, followed by conclusions and discussion of future work in Section 5.

2 Related Work

2.1 Task Oriented Conversational Systems

Recently, end-to-end approaches for dialogue modeling, which use seq2seq-based models, have shown promising

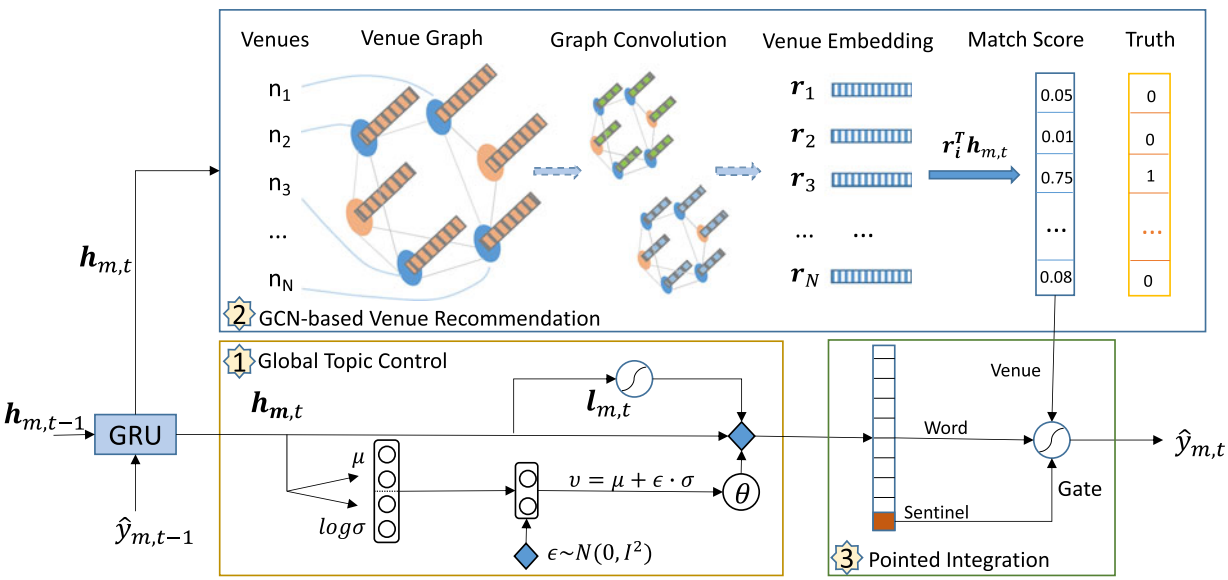

Fig. 2. The proposed TCR model. The global topic component enables the system to switch between various sub-tasks quickly. The GCN-based component generates venues by considering various information and relations. Finally, a pointed integration mechanism incorporates the two for final response generation. The diamonds are stochastic nodes.

results [6], [8], [15]. They directly map plain text dialogue history to the output responses. Since the dialogue states are latent, there is no need for hand-crafted state labels. In order to make such models generate within-topic responses, a possible way is to provide relevant database query results as a proxy for language grounding. As shown in [16], a stochastic neural dialogue model can generate diverse yet rational responses mainly because they are heavily driven by the knowledge the model is conditioned on. However, despite the need for explicit knowledge representations, building a corresponding knowledge base and actually making use of it have been proven difficult [17], [18]. Therefore, progress has been made in conditioning the seq2seq model on coarse-grained knowledge representations, such as a fuzzily-matched retrieval result via attention [19] or a set of pre-organized topic or scenario labels [20], [21]. In our work, we opt for a new direction to employ a hybrid of a seq2seq conversational model and a neural topic model to jointly learn the useful latent representations. Based on the learned topics, the system manages to narrow down the response generation.

2.2 Conversational Recommender

By offering a natural way for product or service seeking, conversational recommendation systems are attracting increasing attention [1], [2], [3]. Due to the big commercial potential, companies like Amazon, Google, eBay, Alibaba are all rolling out such kind of conversational recommenders. Intuitively, integrating recommendation techniques into conversational systems can benefit both recommender and conversational systems, especially for travel. For conversational systems, good venue recommendations based on users' utterances, venue information and relations can better fulfill user's information need thus creating more business opportunities. For recommender systems, conversational systems can provide more information about user intentions, such as user preferred type of food or the location of a hotel, by interactively soliciting and identifying user intentions based on multi-round natural language conversation.

Although conversational recommendation has shown great potential, research in this area is still at its infancy. Existing approaches usually are goal-oriented and combine various modules each designed and trained independently [22], [23], [24]. These approaches either rely heavily on tracking the dialogue state which consists of slot-value pairs, or focus on different objectives such as minimizing the number of user queries to obtain good recommendation results. For example, [12] employed user-based autoencoder for collaborative filtering and pre-trained it with MovieLens data to do recommendation. However, their recommendations are only conditioned on the movies mentioned in the same dialogue, while ignores other dialogue contents expressed in natural language. As another example, [25] leveraged a generative Gaussian model to recommend items to users in a conversation. However, their dialogue system only asks questions about whether a user likes an item or whether the user prefers an item to another, while a typical task oriented dialogue system often directly solicits facets from users [8], [26]. Therefore, [1] defined a new system ask –user respond paradigm for conversational search. [2] designed a new approach to obtaining user preferences in dialogue and contributed a large dataset. [3] proposed to interactively recommend a list of items with visual appearance to harvest more effective user feedback. There are also another line of approaches using reinforcement learning (RL) to train goal-oriented dialogue systems [11], [27]. For instance, in [13], a simulated user is used to help train a dialogue agent to extract the facet values needed to make an appropriate recommendation. In contrast, we propose to employ a GCN-based venue recommender to take care of various relations for venues and seamlessly integrate these results to the response generation. From this angle, our work is also related to [28] which stimulates the propagation of user preferences over the set of knowledge entities. However, this method focused on the traditional recommendation scenario.

3 The TCR Model

The complete architecture of our approach is illustrated in Fig. 2. Starting from the bottom of Fig. 2, there are mainly three sub-components as follows.

(1) To help the system generate within-topic response y_m, a global topic control component takes in dialogue context $\{u_1, \ldots, u_{m-1}\}$ and produces probability distribution $\mathbf{p}(y_{mt})$ over each token y_{mt} that favor certain topics

$$\mathbf{p}(y_{mt}) = f_{Topic}(\{u_1, \ldots, u_{m-1}\}|\mathbf{\Psi}),$$

where f_{Topic} denotes the global topic control model network and $\mathbf{\Psi}$ denotes the network parameters.

(2) A graph convolutional neural network based venue recommendation component learns venue representation $\mathbf{R}$ by capturing various venue information and relationships. It learns the matching between dialogue contexts $\{u_1, \ldots, u_{m-1}\}$ and the representations $\mathbf{R}$ to generate recommendation scores $\mathbf{p}$ for venues

$$\mathbf{p} = softmax(\mathbf{R}^T\mathbf{h}),$$

where $\mathbf{h}$ is the hidden representation of dialogue context.

(3) The recommender's output $\mathbf{p}$ is used in response generation together with the topic part output $\mathbf{p}(y_{mt})$ via a pointed integration mechanism. The hard gate sentinel \$ is leveraged for choosing them.

3.1 Global Topic Control

3.1.1 Basic Encoder

Formally, we consider a dialogue as a sequence of M utterances $D = \{u_1, \ldots, u_M\}$. Each utterance u_m is a sequence with N_m tokens, i.e., $u_m = \{y_{m,1}, \ldots, y_{m,N_m}\}$. The $y_{m,n}$ are either tokens from a vocabulary V or venue names from a set of venues V'. In general, seq2seq-based conversational models like [6] generate a target utterance given a source utterance and dialogue history. Given the dialogue context $\{u_1, \ldots, u_{m-1}\}$, the goal is to produce a *machine* response u_m that maximizes the conditional probability $u_m^* = argmax_{u_m} p(u_m|u_{m-1}, \ldots, u_1)$. Here, we apply the well-accepted hierarchical recurrent encoder decoder (HRED) model [6] as the backbone network. At the token level, an encoder RNN maps each utterance u_m to an utterance vector representation $\mathbf{u}_m$, which is the hidden state obtained after the last token of the utterance has been

processed. At the utterance level, a context RNN keeps track of past utterances by iteratively processing each utterance vector and generates the hidden state $\mathbf{h}_m$

$$p(\mathbf{u}_m|\mathbf{u}_{m-1},\ldots,\mathbf{u}_1) \triangleq p(\mathbf{u}_m|\mathbf{h}_m) \tag{1}$$

$$\mathbf{h}_m = f_{\mathbf{W}_U}(\mathbf{h}_{m-1},\mathbf{u}_{m-1}). \tag{2}$$

At the token level, when the decoder of the HRED model generates tokens in *machine* response u_m, we initialize $\mathbf{h}_{m,0} = \mathbf{h}_{m-1}$

$$p(\mathbf{y}_{m,t}|\mathbf{y}_{m,1:t-1},\mathbf{h}_{m-1}) \triangleq p(\mathbf{y}_{m,t}|\mathbf{h}_{m,t}) \tag{3}$$

$$\mathbf{h}_{m,t} = f_{\mathbf{W}_H}(\mathbf{h}_{m,t-1},\mathbf{y}_{m,t-1}), \tag{4}$$

where $\mathbf{h}_{m,t}$ is the token level hidden state at step t inside turn m, $f_{\mathbf{W}_U}$ and $f_{\mathbf{W}_H}$ are the hidden states that can either be a vanilla RNN cell or complex cell like LSTM or GRU.

3.1.2 Generative Process

While RNN-based models can theoretically model arbitrarily long dialogue histories if provided enough capacity, in practice even the improved version like LSTM or GRU struggles to do so [29], [30]. In dialogues between user and agent, there usually exist long-range dependencies captured by topics such as hotel reservation, restaurant finding and train ticket booking etc. Since much of the long-range dependency in language comes from semantic coherence [30], not from syntactic structure which is more of a local phenomenon, the inability to memorize long-term dependencies prevents RNN-based models from generating within-topic responses. On the other hand, topic models are a family of models that can be used to capture global semantic coherency [14]. It relies on counting word co-occurrence to group words into groups. Therefore, we leverage a neural topic component to extract and map between the input and output global semantics so that the seq2seq submodule can focus on perfecting local dynamics of the utterances such as the syntax and word order.

The generative process of the global topic control component can be described as in Algorithm 1.

Algorithm 1. The Generative Process

1. Encode the user input u_{m-1} and dialogue context C: $\mathbf{h}_{m-1} = HRED(u_{m-1},\ldots,u_1) \in \mathbb{R}^d$.
2. Draw a topic proportion vector $\boldsymbol{\theta} \sim N(0,\mathbf{I})$.
3. In turn m, initialize the decoder hidden state $\mathbf{h}_{m,0} = \mathbf{h}_{m-1}$.
4. Given token $y_{m,1:t-1}$, for the tth token $y_{m,t}$:
   Update the hidden state: $\mathbf{h}_{m,t} = f_{\mathbf{W}_H}(\mathbf{h}_{m,t-1},\mathbf{y}_{m,t-1})$.
   Draw stop word indicator:
   $l_t \sim Bernoulli(sigmoid(\mathbf{W}^T\mathbf{h}_{m,t}))$.
   Draw a token $y_{m,t} \sim p(y_{m,t}|\mathbf{h}_{m,t},\boldsymbol{\theta},l_t,\mathbf{B})$, where
   $p(y_{m,t}=i|\mathbf{h}_{m,t},\boldsymbol{\theta},l_t,\mathbf{B}) \propto exp(\mathbf{w}_i^T\mathbf{h}_{m,t} + (1-l_t)\mathbf{b}_i^T\boldsymbol{\theta})$.

The $HRED(\cdot)$ is the HRED model [6] which encodes dialogue history into a vector representation, and $N(\mu(\mathbf{h}_{m-1}), \sigma^2(\mathbf{h}_{m-1}))$ is a parametric isotropic Gaussian with a mean and variance both obtained from Multilayer Perceptron with input $\mathbf{h}_{m-1}$ separately. The $\mathbf{w}_i$ and $\mathbf{b}_i$ are the corresponding columns in weight matrix $\mathbf{W}$ and $\mathbf{B}$. To combine with the seq2seq-based model, we adopt the hard-decision style from TopicRNN [30] by introducing a random variable l_t. The stop word indicator l_t controls how the topic vector $\boldsymbol{\theta}$ affects the output. Note that the topic vector is used as a bias which enables us to have a clear separation of global semantics and those of local dynamics. For example, when $l_t = 1$ which indicates that $y_{m,t}$ is a stop word, the topic vector $\boldsymbol{\theta}$ will have no contribution to the output. This design is especially useful as topic models do not model stop words well, because stop words usually do not carry semantic meaning while appear frequently in almost every dialogue session.

3.1.3 Inference

During model inference, the observations are token sequences u_m and stop word indicators $l_{1:N_m}$. The log marginal likelihood of u_m is

$$\begin{aligned} &log\ p(u_m, l_{1:N_m}|u_{1:m-1}) \\ &= log \int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|u_{1:m-1}) \prod_{t=1}^{N_m} p(y_{m,t}|\mathbf{h}_{m,t},l_t,\boldsymbol{\theta}) p(l_t|\mathbf{h}_{m,t}) d\boldsymbol{\theta}. \end{aligned} \tag{5}$$

Since direct optimization of Eq. (5) is intractable due to the integral over the continuous latent space, we use variational inference for approximating it [31]. Suppose $q(\boldsymbol{\theta}|u_{1:m})$ be the variational distribution on the marginalized variable $\boldsymbol{\theta}$, the variational lower bound of Eq. (5) can therefore be constructed as

$$\begin{aligned} &\mathcal{L}(u_m, l_{1:N_m}|q(\boldsymbol{\theta}|u_{1:m}),\boldsymbol{\Psi}) \\ &\triangleq \mathbb{E}_{q(\boldsymbol{\theta}|u_{1:m})}\Big[\sum_{t=1}^{N_m} logp(y_{m,t}|\mathbf{h}_{m,t},l_t,\boldsymbol{\theta}) \\ &+ \sum_{t=1}^{N_m} logp(l_t|\mathbf{h}_{m,t})\Big] - D_{KL}(q(\boldsymbol{\theta}|u_{1:m})||p(\boldsymbol{\theta}|u_{1:m-1})) \\ &\leq log\ p(u_m, l_{1:N_m}|u_{1:m-1},\boldsymbol{\Psi}). \end{aligned} \tag{6}$$

Inspired by the neural variational inference framework in [32], [33] and the Gaussian reparameterization trick in [34], we construct $q(\boldsymbol{\theta}|u_{1:m})$ as an inference network using a feed-forward neural network

$$q(\boldsymbol{\theta}|u_{1:m}) = N(\boldsymbol{\theta};\mu(u_{1:m}), diag(\sigma^2(u_{1:m}))). \tag{7}$$

Denoting $\boldsymbol{\tau} \in \mathcal{N}_+^{|V/V_s|}$ as the term-frequency vector of $u_{1:m}$ excluding stop words (with V_s as the stop word vocabulary), we have $\mu(u_{1:m}) = ReLU(\mathbf{W}_\mu^T\boldsymbol{\tau})$ and $\sigma(u_{1:m}) = ReLU(\mathbf{W}_\sigma^T\boldsymbol{\tau})$ where bias is omitted. Note that although $q(\boldsymbol{\theta}|u_{1:m})$ and $p(\boldsymbol{\theta}|u_{1:m-1})$ are both parameterized as Gaussian distributions, the former one only works during training while the later one generates the required topic distribution vector $\boldsymbol{\theta}$ for composing the machine response.

Suppose during training, the one-hot vector for any token y and its stop word indicator are $\mathbf{y}$ and $\mathbf{l}$ respectively. The predicted correspondence vectors are $\mathbf{y}'$ and $\mathbf{l}'$. Inspired by Eq. (6), the loss for this global topic control component consists of two cross entropy losses and a KL divergence between the assumed distribution and learned distribution as follows:

$$\mathcal{L}_{Topic} = \ avg.\ \Big[\mathcal{L}_{cross}(\mathbf{y}, \mathbf{y}') + \mathcal{L}_{cross}(\mathbf{l}, \mathbf{l}')\Big] - D_{KL}(N(0, \mathbf{I})||q(\boldsymbol{\theta}|u_{1:m})), \quad (8)$$

where $avg.$ indicates the averaged cross entropy loss over all training tokens.

3.2 GCN-Based Venue Recommendation

Given the dialogue context and ground truth venue node pairs, our task in this subsection is to find a good match between them. We need to leverage both the venue attributes such as 'free wifi' for hotel and the various relationships between these venues. For example, when user books a hotel, he or she might also want to find a 'nearby' restaurant. To jointly consider such attributes as well as the relationships, we naturally resort to graph based methods. Recently, the graph convolutional neural network (GCN) based methods have set a new standard on countless recommender system benchmarks [35], [36]. Unlike purely content-based deep models (e.g., recurrent neural networks), GCNs leverage both content information as well as graph structure. We thus adopt the graph convolution operation into our venue recommender.

We formulate an un-directed graph structure as $G = (O, E)$, where $O = \{n_1, n_2, \ldots, n_N\}$ is a set of N nodes and $E \subseteq N \times N$ is a set of edges between nodes. Note that we model two kinds of nodes in the graph. One is venue entities such as hotel names, restaurant names or train numbers etc, the other one is venue attribute nodes or say slot value nodes, such as values of location, price range *etc*. For example, the location slot has five values: north, east, center, south, west of the city which yields five venue attribute nodes in the graph. The slot values come from the slot ontology[2] defined in the dialogue dataset. In general, we selected 214 venue entities and 59 slot values to construct the final graph.

Regarding the edges in the graph, the venue attribute nodes are mainly used for connecting venues thus edges are formed between venue entity nodes and their corresponding attribute nodes. For venues co-occurred in the same dialogue session, we also construct edges between them. Finally, there are 674 edges in the graph and most of them are constructed via slot value belonging relationship. We use $\mathbf{A} \in \mathbb{R}^{N \times N}$ to denote the adjacency matrix, $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ to denote the adjacency matrix with added self-connections and the new degree matrix $\widetilde{\mathbf{D}}_{ii} = \sum_j \widetilde{\mathbf{A}}_{ij}$. Since there are also some brief description about venues provided in dataset, we leverage such information to build an initial information matrix $\mathbf{X}$ for venues. We denote the representations of nodes in lth layer as $\mathbf{R}^{(l)}$. Initially, we have $\mathbf{R}^{(0)} = \mathbf{X}$.

An important issue we need to point out here is that, though we model both venue entities and slot values in the graph, only the representations learned for venue entities are leveraged for dialogue history matching. The slot value nodes are actually treated as latent in the learning procedure. We tried to directly connect venue entities by edges while ignore slot value nodes which result in much lower performance thus we discarded this way.

Given such a constructed graph, we generate high-quality embeddings or representations of entities that can be used for calculating the matching score with dialogue context thus obtaining the venue recommendation results. Generally speaking, to generate the embedding for a venue, we apply multiple convolutional modules that aggregate feature information from the venue's local graph neighborhood. The core idea is to learn how to iteratively aggregate feature information from local graph neighborhoods. As shown in Fig. 3, we first project the former layer node representation $\mathbf{R}^{(l-1)}$ into a latent space using the weight matrix $\mathbf{W}^{(l)}$ (we omitted the bias term for simplicity)

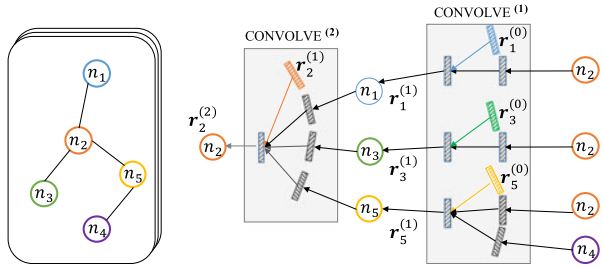

Fig. 3. The illustration of convolution operation in the constructed graph. Two layers are stacked. Each $\mathbf{r}^{(l)}$ denotes a node representation, corresponds to the column in $\mathbf{R}^l$.

$$\mathbf{R}' = \mathbf{R}^{(l-1)}\mathbf{W}^{(l)}.$$

Then the latent representation $\mathbf{R}'$ is propagated via the normalized adjacency matrix $\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}$ with self-connections. As demonstrated in [37], this propagation rule is motivated via a first-order approximation of localized spectral filters on graphs. Finally, we use the ReLU function to increase the non-linearity. Thus, a single "convolution" operation transforms and aggregates feature information from a node's one-hop graph neighborhood as follows:

$$\mathbf{R}^l = ReLU(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{R}^{(l-1)}\mathbf{W}^{(l)}). \quad (9)$$

By stacking multiple such convolutions, information can be propagated across far reaches of a graph. Here we stack two layers.

After introducing the updating rules for node representations as in Eq. (9), we present the objective function which encourages the matching between dialogue context and venues. Suppose there are M dialogue context and ground truth node pairs, we obtain the dialogue context representation $\mathbf{h}_i$ and the ground truth node vector $\mathbf{s}_i \in \mathbb{R}^N$ for each pair. The objective function resumes the cross-entropy loss as follows:

$$\mathcal{L}_{GCN} = -\frac{1}{M}\sum_{i=1}^{M}[\mathbf{s}_i log(\mathbf{p}_i) + (1 - \mathbf{s}_i) log(1 - \mathbf{p}_i)], \quad (10)$$

where $\mathbf{p}_i = softmax(\mathbf{R}^T\mathbf{h}_i)$ is a vector of scores predicted by the GCN-based model, and $\mathbf{R}$ is the finial node representation matrix obtained via the graph convolution process.

3.3 Pointed Integration Mechanism

Now, given the dialogue context, we can predict the next utterance via the global topic control component (*branch 1*) and obtain the recommended venue through the GCN-based recommender (*branch 2*). We sum up the process described in former subsections here: at each decoding step

2. We ignore slots such as number-of-people, stay, day *etc*.

t in turn m, the hidden state $\mathbf{h}_{m,t}$ is passed to the two branches as shown in Fig. 2.

In branch 1, the $\mathbf{h}_{m,t}$ is passed to the global topic control component. Following the generative process introduced in Section 3.1, the probability of generating the next token is calculated as

$$\mathbf{p}_1(\hat{y}_{m,t}) \propto exp(\mathbf{W}^T\mathbf{h}_{m,t} + (1-l_t)\mathbf{B}^T\boldsymbol{\theta}). \tag{11}$$

In branch 2, the $\mathbf{h}_{m,t}$ is fed to the GCN-based recommender. Following the process introduced in Section 3.2, the recommender ranks the venues and outputs the top ranked venue name

$$\mathbf{p}_2(\hat{y}_{m,t}) = softmax(\mathbf{R}^T\mathbf{h}_{m,t}). \tag{12}$$

To integrate the two lines of results, we propose a pointed integration mechanism. In the final response generation, whether a token is generated from Eqs. (11) or (12) is decided via a sentinel. As detailed before, we have a set of venue names V'. At the very beginning, we substitute all the venue names in dataset with the sentinel token \$. Thus the vocabulary for topic control component is V which consists of all the tokens appearing in our dataset (expect the venue names) plus the \$ token. During the response decoding process, once the sentinel is chosen, the model will generate the token from the GCN-based recommender, which means the model will produce the top-ranked venue name as the generated token (i.e., choose the branch of Eq. (12)). Otherwise, the model chooses a token in V as the decoded token (i.e., choose the branch of Eq. (11)). Basically, similar to [38], the sentinel token is used as a hard gate to control where the next token is generated from at each time step. In this way, we do not need to separately learn a gating function as in [39]. Also, our model is not constrained by a soft gate mechanism as in [40].

3.4 Training Objectives

As the generation of responses is controlled via the sentinel token \$ as a hard gate, the generation procedure actually works in a two-step way. The substitution of \$ with venue recommendation result is separate from the token generation process. In order to achieve good results, we train the whole model in a sequential way. At the beginning, we train the global topic control component separately on the altered dataset where all venue names are replaced with \$. The training objective of this component is $\mathcal{L}_{Topic}$ detailed as Eq. (8).

Then we change back the dataset and train the GCN component for venue ranking on it. The dialogue context is embedded via the trained global topic control model. The training objective is $\lambda\mathcal{L}_{GCN}$ as detailed in Eq. (10).

Finally, we initialize the whole model with the components trained and fine-tune them altogether. The final training objective is as follows:

$$\mathcal{L} = \mathcal{L}_{Topic} + \lambda\mathcal{L}_{GCN},$$

where λ is the weight to balance the losses of the two components. In our experiments, we empirically set this hyperparameter to 0.1.

4 Experiments

In this section, we systematically evaluate the proposed method, termed as *TCR*. The experiments are carried out to answer the research questions as follows.

RQ1: Can the proposed TCR properly respond to users' queries in multi-domain? What are the key reasons behind?

RQ2: Does the topic control component help the system generate coherent responses? Are the learnt topics reasonable?

RQ3: Does the GCN-based recommender help the system find appropriate venues? Whether the relationships between venues are important to capture?

4.1 Experimental Setup

4.1.1 Dataset

Arguably the greatest bottleneck for statistical approaches to dialogue system development is the collection of appropriate training dataset, and this is especially true for task-oriented dialogue systems [41]. Fortunately, [9] contributed a dataset consisting of over 10K conversation sessions in multi-domain — MultiWOZ, which is a fully-labeled collection of human-human written conversations. During the collection of this dataset, it simulates natural conversations between a tourist and a clerk from an information center in a touristic city. Various possible dialogue scenarios are considered, ranging from requesting basic information about attractions through booking a hotel room or traveling between cities. In total, the presented corpus consists of 7 domains — Attraction, Hospital, Police, Hotel, Restaurant, Taxi, Train. The dialogues cover between 1 and 5 sub-topics per dialogue thus greatly varying in length and complexity. This broad range of topics captures the common scenarios where tasks are naturally connected in travel. For example, a tourist needs to find a hotel, to get the list of attractions and to book a taxi to travel between both places.

In total, there are 10,438 dialogues collected, where 3,406 of them focus in single-topic dialogues and 7,032 of them are dialogues consisting of at least 2 up to 5 sub-topics. In the experiment, we follow random split of train, test and development set in the original paper. The test and development sets contain 1k examples each. Generally speaking, around 70 percent of dialogues have more than 10 turns which shows the complexity of the corpus. The average number of turns are 8.93 and 15.39 for single and multi-domain dialogues respectively with 115, 434 turns in total. The average sentence lengths are 11.75 and 15.12 for users and system response respectively. The responses are also more diverse thus enabling the training of more complex generation models.

4.1.2 Comparing Methods

To evaluate the effectiveness of the proposed method, we compare it with the following state-of-the-art solutions.

HRED [6]: It predicts the system utterance given the history utterances. The history is modeled with two RNNs in two levels: a sequence of tokens for each utterance and

a sequence of utterances. This model works as the basis for our method and other baselines.

MultiWOZ [9]: It frames the dialogue as a context to response mapping problem, a seq2seq model is augmented with an oracle belief tracker and a discrete database accessing component as additional features to inform the word decisions in the decoder. Note that a seq2seq model is used in the original paper, we extend it to HRED to model multi-turn dialogues.

Mem2Seq [38]: It augments the existing MemNN [42] framework with a sequential generative architecture to produce coherent responses for task-oriented dialogue systems. It uses global multi-hop attention mechanisms to copy words directly from dialogue history or KBs.

TopicRNN [30]: It incorporates topic information into the seq2seq framework to generate informative and interesting responses for chatbots. We also extend the encoder part to model multi-turn dialogues.

ReDial [12]: It integrates the HRED based conversational model with a denoising auto-encoder based recommender [43] via a switching mechanism. The recommendation part is pre-trained separately and only considers the co-occurrence of items while ignores the dialogue context. The recommender part is also compared in ablation study.

NCF [44]: It employs deep learning to model the key factor in collaborative filtering — the interaction between user and item features, and achieves good performance. The inner product is replaced with a neural architecture. We compare this recommender with our GCN-based recommender in the ablation study.

RippleNet [28]: It stimulates the propagation of user preferences over the set of knowledge entities by iteratively extending a user's potential interests along links in the knowledge graph. We compare this recommender with our GCN-based recommender in the ablation study.

4.1.3 Evaluation Protocols

We evaluate the methods in various evaluation protocols. Due to the difficulty in evaluating conversational agents [45], a human evaluation is usually necessary to assess the performance of the models. Therefore, we perform both corpus-based evaluations and human evaluations. For corpus-based evaluations, we adopt the BLEU score and Entity Accuracy as our evaluation metrics, where:

- *BLEU*: Being commonly used in machine translation evaluations, BLEU score has also been widely used in evaluating dialogues systems [46]. It is based on the idea of modified n-gram precision, where the higher score denotes better performance.
- *Entity Accuracy*: Similar to [38], we average over the entire set of system responses and compare the entities in plain text. The entities in each system response are selected by a predefined entity list. This metric evaluates the ability to recommend items from the provided item set and to capture the semantics of dialogues [46].

For human evaluations, we define a set of subjective scores to evaluate the performance of various methods. We run a user study to assess the overall quality of the responses of our model as compared to the baselines. To do a less biased evaluation, we recruit five participants (both are graduate students: four males and one female student). We present each of them ten generated dialogue sessions from our test set. The participants are asked to give Fluency scores and Informativeness scores for the generated system responses. They are also asked to provide the rankings of each method for each dialogue session. We allow ties so that multiple methods could be given the same rank for the same dialogue session (e.g., rankings of the form 1, 2, 2, 2, 2 are possible if the one method is clearly the best, but the other four are of equivalent quality).

- *Fluency*: It evaluates how fluent the generated responses are. The score ranges from zero to five, where a larger score indicates the generated response is more fluent.
- *Informativeness*: This score shows whether the generated responses are informative or not, or say whether users' queries get properly answered. It also ranges from zero to five, where a larger score indicates that the evaluator thinks that the generated response is more informative.
- *Ranking*: This metric directly shows how good each method is as compared to the others. It reflects the overall feeling of users regarding the performance of each method.

4.1.4 Training Details

The proposed model is implemented in PyTorch. We use the provided development set to tune the hyper-parameters, track the training progress and select the best performing model for reporting the results on the test sets. The components of the joint architecture are first trained separately to achieve a relatively good performance. We then combine them together and fine-tune by minimizing the sum of various loss functions as detailed in Section 3.4. We use an embedding size of 300, GRU state size of 100. The embeddings are initialized from pre-trained GloVe embeddings [47] and fine-tuned during training. We use two layers of graph convolutional operations. Mini-batch SGD with a batch size of 64 and Adam optimizer with a learning rate of 0.01 are used for training.

We use the Python-based natural language toolkit NLTK to perform tokenization. Entities in dialogue sessions are recognized via heuristic rules plus database entries. All counts, time and reference numbers are replaced with the $< \text{value\_count} >$, $< \text{value\_time} >$ and $< \text{domain\_reference} >$ tokens respectively. To reduce data sparsity further, all tokens are transformed to lowercase letters. The stop words are chosen using tf–idf [48]. The number of topics K is set to 20. All tokens that appear less than 5 times in the corpus are replaced with the $< \text{UNK} >$ token. We follow the {S,U,S'} utterance "triples" structure as [6] in our experiments, which means we aim to generate the system utterance S' by observing the former 1 turn of system utterance S and user utterance U.

4.2 Performance Comparison (RQ1)

4.2.1 Corpus-Based Evaluation

The result of the corpus-based evaluation is presented in Figs. 4 and 5. For each method, the results are obtained

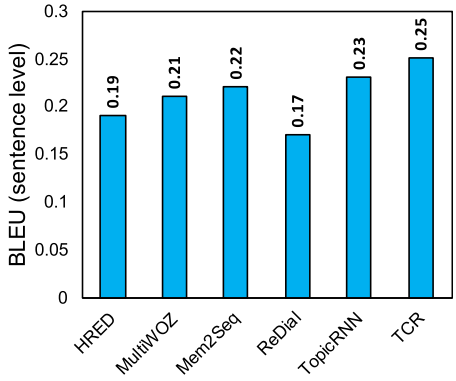

Fig. 4. The BLEU scores for each method. (RQ1).

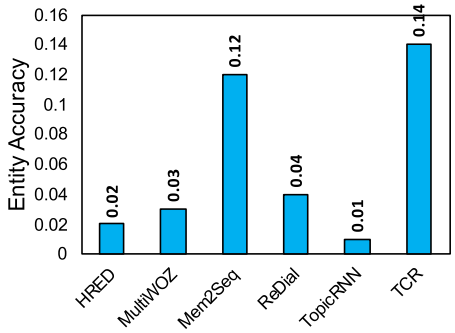

Fig. 5. The entity accuracy scores for each method. Note that this is the top-1 accuracy score since only the top ranked venue is leveraged by text response. (RQ1).

based on the best model chosen via the development set. The key observations are as follows.

Overall, the proposed TCR method performs better than all the other baselines in both metrics – BLEU and entity accuracy. For example, regarding BLEU score, we observe a 6.82 percent of performance improvement as compared to the second best method, TopicRNN. The two methods perform better than all the other conversational baselines. In terms of the entity accuracy score, TCR improves the performance of venue recommendation by 17.2 percent as compared to the second best method, Mem2Seq. The performance improvements of TCR method demonstrate its effectiveness in multi-domain conversational recommendation due to the following aspects: a) TCR has a global topic control component which enables the system to adaptively generate within topic responses based on the context topic. The learned topics narrow down the generation of tokens in decoding. b) The graph convolution operation incorporates venue information as well as venue relations in the learned venue representations. It matches the venues with the dialogue contexts which is essential for conversational recommendation.

In more detail, we analyse the BLEU score shown in Fig. 4 first. It reflects the quality of generated text responses. Generally speaking, all methods manage to achieve some improvements over the basic framework – HRED. For the MultiWoz method, the performance improvement is due to the incorporation of a belief tracker and a discrete database accessing component. However, the improvement is less than that of the Mem2Seq method, because MultiWoz encodes the belief states into anonymous vectors and only the database search count is leveraged. Mem2Seq, on the contrary, generates responses from the dialogue history and KB — some tokens or entities are directly copied to form responses. It happens frequently that words appeared in dialogue context are re–used by later responses, which is the underlying reason for its good performance. For the method ReDial, since a pointer softmax is leveraged to integrate the text modeling and the recommendation part, its BLEU score might get affected. When it comes to TopicRNN, we observe a performance improvement, which is mainly attributed to the topic mechanism. It helps to generate tokens matching the dialogue context topic and narrow down the generation of tokens. In addition to a similar topic control scheme, TCR manages to achieve superior performance by achieving better entity prediction.

Regarding the entity accuracy score presented in Fig. 5, we observe that the basic end-to-end framework method, HRED, performs rather badly. It is as expected since the method only treat venue entities as tokens and generate tokens based on the encoded dialogue context. The basic information of venue entities and the relationships between them are ignored. For the MultiWoz method, although a database query component is leveraged, it only makes use of the number of obtained results. Therefore, the performance is still relatively low. When it comes to the Mem2Seq method, there is a large performance leap. By observing the corpus, we find that the reason might be due to frequent entity re-use phenomenon in dialogues as we detailed before — venue entities appeared in dialogue context will likely to re-appear in the following responses. For the ReDial method, it manages to achieve better performance than that of its basic framework HRED but the improvement is limited. Although it has a denoising autoencoder based recommender, it is largely affected by the data sparsity problem in the dataset, and the recommendations are only conditioned on the entities mentioned in the context but not directly on the language, e.g., texts like "a cheap restaurant" in dialogue context are ignored. For the TopicRNN method, we also observe a rather low performance on entity prediction. The reason behind is similar to that of the HRED method. On the contrary, the proposed TCR method is able to achieve superior performance on finding the appropriate venue entities. This is because the GCN-based recommender jointly considers the venue information, venue relationships and their match to the dialogue context.

4.2.2 Human Evaluation

We present the averaged human evaluation results in Table 1 (the Fleiss' kappa value between evaluators is 0.65). It directly reflects human perception of the quality of generated responses. The results show that the proposed TCR achieves

TABLE 1
Human Evaluation Results for Different Methods

Method	Fluency	Informativeness	Ranking
HRED	2.64	2.34	3.08
MultiWOZ	2.74	2.82	2.7
Mem2Seq	3.04	3.06	2.3
ReDial	2.58	2.62	2.8
TopicRNN	3.64	2.78	2.66
TCR	**3.96**	**3.82**	**1.8**

(RQ1)

TABLE 2
Representative Topics From the Global Topic Control Component

Restaurant	Hotel	Attraction	Taxi
restaurant	hamilton	region	runs
eastern	guesthouse	shopping	vehicle
cantonese	convenient	modern	departures
appeal	stayed	fabulous	campus
vegetarian	aylesbray	world	birmingham
menu	warkworth	churchhill	arriveby
eritrean	accommodation	christ	driving
caribbean	arrangements	shopping	causeway

(RQ2)

the best performance across these various metrics, which indicates that the responses generated by it are more fluent and informative. We show that the performance improvements of TCR over the other methods are significant. For example, in terms of the Fluency score, TCR improves the performance of response generation by 50.0, 44.5, 30.3, 53.5 and 8.8 percent as compared to the HRED, MultiWoz, Mem2Seq, Redial and TopicRNN methods, respectively. Intuitively, at a certain degree, the BLEU score also reflects how fluent the responses are. In the results, these two metrics indeed show similar pattern of performance improvements. As detailed before, the main reason for the superior performance of TCR might be due to the global topic control mechanism. In real life scenarios, intelligent systems naturally involve solving multiple tasks, which leads to several topics in the dialogue flow. The topic control component enables the system to swiftly switch among topics and generate within-topic responses.

At the same time, the Informativeness score shows whether user queries are properly addressed. It not only includes the evaluation of recommended venues but also the information slots appeared in responses such as *food type*, *hotel price* etc. We observe that the general performance pattern resembles that of the entity accuracy metric. However, the Informativeness score of TCR is much larger than that of Mem2Seq. This might be due to the fact that although the venue entities can re-occur in responses, the value of information slots usually require venue specific knowledge.

For the final ranking of methods, we find it in general accordance with the Fluency and Informativeness score trends. The TCR is ranked as the best method, followed by the Mem2Seq method. It actually points out a future direction to enhance our method – to make good use of the "re-use" phenomenon. The reason behind is due to the interactive and updating nature of dialogues. In the course of dialogue, the user and agent interact to reach the same level of information awareness regarding a specific task. Consequently, during the interaction process, many words or phrases will be inevitably repeated by them during question answering, requirement modification or confirmation *etc*. Therefore, to encode dialogue history and decode response directly via HRED might not be sufficient. Incorporating copy mechanism to give special treatment for tokens from history as in the Mem2Seq method opens a new auxiliary road for response generation in dialogues.

4.3 Analysis on Components

In this subsection, we explore the performance and contribution of the major components in our design.

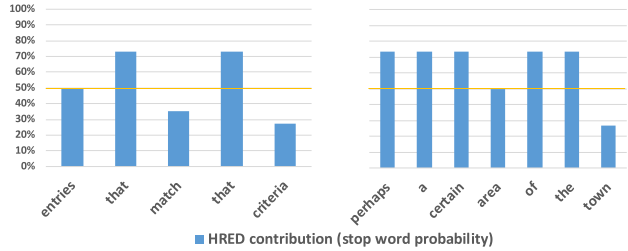

Fig. 6. Analysis of the learned stop word indicators. (RQ2).

4.3.1 Topic Control of Dialogues (RQ2)

Here we evaluate the performance of the global topic control component. First of all, we test the performance of removing the global topic control component while just using the HRED. The BLEU score degrades to be similar to that of the HRED baseline. This is as expected, since the encoder and decoder are generally the same now while the effect from GCN part is rather limited due to our hard gate mechanism. For the entity accuracy, the score becomes about 13 percent lower than that of TCR. It is still better than that of Mem2Seq, but is much better than that of HRED. It demonstrates that the major performance gain comes from the effectiveness of GCN in capturing venue and attribute relations, and the topic control exerts relatively limited effect on recommendation during the decoding process.

Also, we test whether the learned topic words are coherent. We run the component on dataset with the total topic number K set to 20. To give a clear view, we show several representative topic words in Table 2. The first row entries indicate the estimated topics for their corresponding column of topic words, where these topic words are top-ranked ones within each column group. Generally speaking, we observe that words are grouped together and the top-ranked words show certain topic meanings within each group.

To further show whether the proposed model can coordinate the contribution of the global topic part and the local syntactic, we further analyze the probability of words being assigned to stop words. This is because many words in utterances are actually stop words which do not belong to any topic. The generation of stop words are mainly controlled by HRED locally and can largely affect the performance of the final model. Therefore, we plotted the HRED contribution ratios of some sample words in utterances as in Fig. 6. Basically, we observed that the probability of words like 'a', 'of', 'the' are relatively high which is desired as these are common stop words. Other words such as 'town', 'criteria' are controlled by topic models more thus the HRED rate is relatively small, which is also desired.

To show whether the global topic control component correctly captures the overall topic distribution of dialogue sessions, we plot the topic distribution θ of dialogues as shown in Fig. 7. Two example dialogue sessions are presented, and both of them are paired with the learned topic distribution. In the first example, the user asks the agent to book a table in a cheap Indian restaurant first, then to recommend an attraction in the center of town. There are two tasks involved during the dialogue session. Accordingly, the global topic control component picks up two topics successfully. By observing the top-ranked topic words for each topic indicators, we find that the picked topic 9 refers to

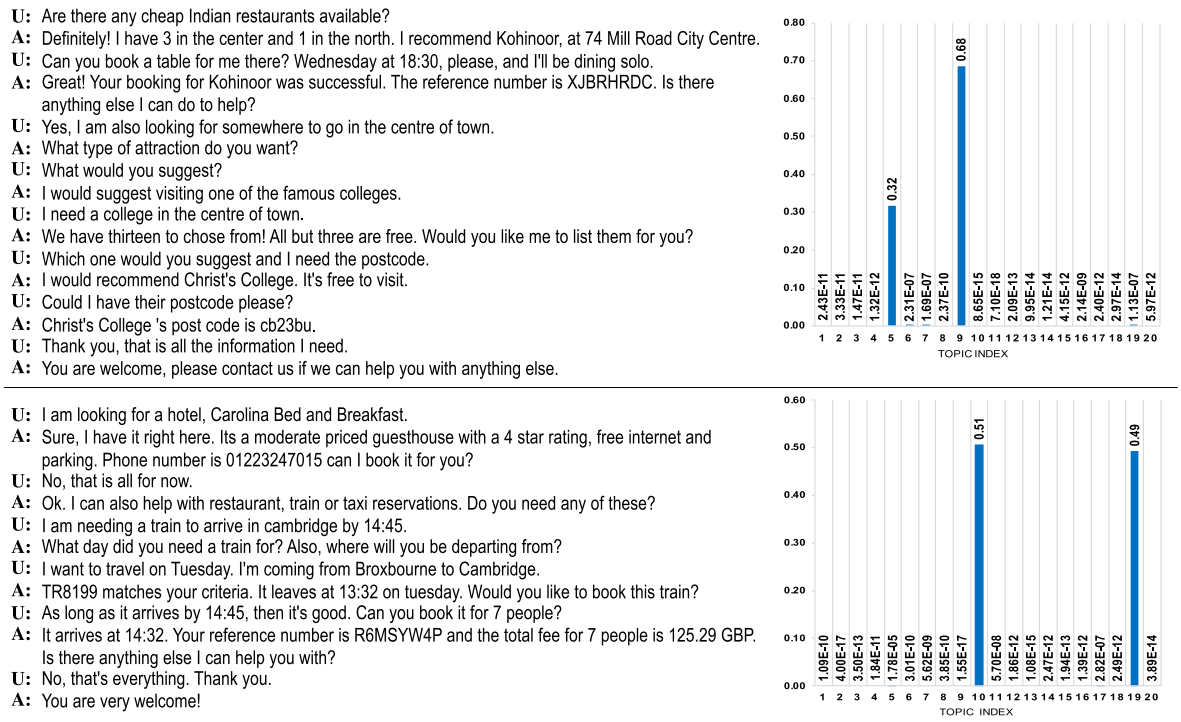

Fig. 7. Inferred topic distribution of two example dialogue sessions. It shows that some of the topics have been picked up depending on the dialogue content. (RQ2).

restaurant while topic 5 refers to *attraction*. It demonstrates the effectiveness of our topic control. Similarly, in the second example, the user first wants to find a hotel and then book the train tickets. There are also two sub-tasks involved in the dialogue. Thus, two topics are picked up by the model. We observe that topic 10 corresponds to *hotel* and topic 19 corresponds to *train*. Indeed, when we delve into individual words, we observe that co-appeared words might belong to different topics which would possibly introduce some noise. However, since the process is probabilistic and each word can be assigned to different topics with different probabilities at the same time, the response fusion process is affected only slightly.

4.3.2 Venue Recommendation Analysis (RQ3)

In this subsection, we analyze our GCN-based venue recommendation component in detail. To test the need of recommenders, we solicit user requirements from utterances using templates and form database queries (DB-Q) to retrieve venues. The results are reported in Table 3. For recommenders, the common user-item interaction situation is abstracted from the dialogues by treating the dialogue contexts as representations of users and venue entities as items. This scenario is used in the NCF method. For ReDial, a user-based autoencoder for collaborative filtering (U-Autorec) is leveraged where venue entities appeared in the same dialogue session are extracted to form the entity vector.

TABLE 3
Performance Comparison of Recommenders

Methods	DB-Q	ReDial	NCF	RippleNet	GCN-based
Top-1 Acc	0.089	0.107	0.188	0.223	**0.242**

Table 3 shows that the GCN-based recommender component achieves better performance as compared to DB-Q, ReDial, RippleNet and NCF methods. The low performance of DB-Q validates the requirement of recommender to handle complicated interactions. For the ReDial recommender, it projects the entity appearance vector v of each dialogue session into a smaller vector space, then retrieve a new entity vector v' with same dimension to minimize the difference between them. It only models the co-occurrence relationship among entities. The entity information and the dialogue context information are largely ignored. At the same time, the entity co-occurrence matrix formed via training dialogue sessions is rather sparse. These factors together lead to its relatively weak performance. Regarding the NCF method, the dialogue contexts are gathered via HRED to form vector representations of users. We adopt a multi-layer perceptron (MLP) to learn the interaction between user and item features. Still, the various relationships between venue entities are not modeled. On the contrary, the GCN-based recommender component in TCR manages to handle all the three evidence sources — the venue information, relations between them and the match to dialogue context. Thus, superior performance is achieved. For RippleNet, the "ripples" are only activated when venues are literally mentioned in the dialogue which might result in insufficient context information thus yield lower performance than the GCN one. Our GCN component captures structural information in database and helps to learn good match between dialogue context and venues, thus manages to achieve better performance.

5 Conclusion

In order to build an intelligent conversational agent that is capable of various tasks across domains, we proposed a deep conversational recommender to answer various user queries.

It is equipped with a global topic control component to adaptively generate within-topic responses based on the dialogue context topics, which narrows down the generation of tokens in decoding. At the same time, a graph convolutional network based recommender manages to pop candidates by modeling the entity information, relations between them and the match to dialogue history. Based on the results from the two components, the final response is generated by incorporating them via a pointed integration mechanism. We systematically evaluated the proposed method on a large-scale multi-domain conversational dataset. Experimental results showed that the proposed TCR method outperformed a wide range of baselines and demonstrated the effectiveness of it in generating fluent and informative responses. In future, we will explore the "re-use" phenomenon to further boost the performance of response generation.

Acknowledgments

This research is part of the NExT++ project, which is supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@SG Funding Initiative.

References

[1] Y. Zhang, 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. Knowl. Manage.*, 2018, pp. 177–186.

[2] F. Radlinski, K. Balog, B. Byrne, and K. Krishnamoorthi, "Coached conversational preference elicitation: A case study in understanding movie preferences," in *Proc. 20th Annu. SIGdial Meet. Discourse Dialogue*, 2019, pp. 353–360.

[3] T. Yu, Y. Shen, and H. Jin, "A visual dialog augmented interactive recommender system," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 157–165.

[4] O. Vinyals and Q. Le, "A neural conversational model," in *Proc. ICML Deep Learn. Workshop*, 2015, pp. 1–7.

[5] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *Proc. 53rd Annu. Meet. Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 1577–1586.

[6] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 3776–3784.

[7] A. Bordes and J. Weston, "Learning end-to-end goal-oriented dialog," in *Proc. 3rd Int. Conf. Learn. Representations*, 2016, pp. 1–14.

[8] T. Wen *et al.*, "A network-based end-to-end trainable task-oriented dialogue system," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 438–449.

[9] P. Budzianowski *et al.*, "MultiWOZ - A large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 5016–5026.

[10] W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, and D. Yin, "Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures," in *Proc. 56th Ann. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1437–1447.

[11] J. Xisen *et al.*, "Explicit state tracking with semi-supervisionfor neural dialogue generation," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manag.*, 2018, pp. 1403–1412.

[12] R. Li, S. E. Kahou, H. Schulz, V. Michalski, L. Charlin, and C. Pal, "Towards deep conversational recommendations," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9748–9758.

[13] Y. Sun and Y. Zhang, "Conversational recommender system," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 235–244.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[15] T. Zhao, A. Lu, K. Lee, and M. Eskenazi, "Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability," in *Proc. 18th Annu. SIGdial Meet. Discourse Dialogue*, 2017, pp. 27–36.

[16] T.-H. Wen, Y. Miao, P. Blunsom, and S. Young, "Latent intention dialogue models," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3732–3741.

[17] C. Matuszek, J. Cabral, M. J. Witbrock, and J. DeOliveira, "An introduction to the syntax and content of Cyc," in *Proc. AAAI Spring Symp.: Formalizing Compiling Background Knowl. Appl. Knowl. Representation Question Answering*, 2006, pp. 44–49.

[18] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2016, pp. 1400–1409.

[19] M. Ghazvininejad *et al.*, "A knowledge-grounded neural conversation model," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5110–5117.

[20] D. Wang, N. Jojic, C. Brockett, and E. Nyberg, "Steering output style and topic in neural response generation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2017, pp. 2140–2150.

[21] C. Xing *et al.*, "Topic aware neural response generation," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3351–3357.

[22] C. A. Thompson, M. H. Goker, and P. Langley, "A personalized system for conversational recommendations," *J. Artif. Intell. Res.*, vol. 21, pp. 393–428, 2004.

[23] C. Greco, A. Suglia, P. Basile, and G. Semeraro, "Converse-Et-Impera: Exploiting deep learning and hierarchical reinforcement learning for conversational recommender systems," in *Proc. Conf. Italian Assoc. Artif. Intell.*, 2017, pp. 372–386.

[24] W. Lei *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.

[25] K. Christakopoulou, F. Radlinski, and K. Hofmann, "Towards conversational recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 815–824.

[26] B. Dhingra *et al.*, "Towards end-to-end reinforcement learning of dialogue agents for information access," in *Proc. 55th Annu. Meet. Assoc. Comput. Linguistics*, 2017, pp. 484–495.

[27] X. Li, Y.-N. Chen, L. Li, J. Gao, and A. Celikyilmaz, "End-to-end task-completion neural dialogue systems," in *Proc. 8th Int. Joint Conf. Natural Lang. Process.*, 2017, pp. 733–743.

[28] H. Wang *et al.*, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 417–426.

[29] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[30] A. B. Dieng, C. Wang, J. Gao, and J. Paisley, "TopicRNN: A recurrent neural network with long-range semantic dependency," in *Proc. 5th Int. Conf. Learn. Representations*, 2017, pp. 1–13.

[31] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, pp. 183–233, 1999.

[32] A. Mnih and K. Gregor, "Neural variational inference and learning in belief networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1791–1799.

[33] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1727–1736.

[34] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Representations*, 2013, pp. 1–14.

[35] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 974–983.

[36] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Eng. Bull.*, pp. 1–14, 2017.

[37] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–14.

[38] A. Madotto, C.-S. Wu, and P. Fung, "Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems," in *Proc. 56th Annu. Meet. Assoc. Comput. Linguistics*, 2018, pp. 1468–1478.

[39] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," in *Proc. 54th Annu. Meet. Assoc. Comput. Linguistics*, 2016, pp. 140–149.

[40] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meet. Assoc. Comput. Linguistics*, 2017, pp. 1073–1083.

[41] L. Liao, Y. Ma, X. He, R. Hong, and T.-S. Chua, "Knowledge-aware multimodal dialogue systems," in *Proc. ACM Multimedia Conf. Multimedia Conf.*, 2018, pp. 801–809.
[42] S. Sukhbaatar *et al.*, "End-to-end memory networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2440–2448.
[43] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 111–112.
[44] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.
[45] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2016, pp. 2122–2132.
[46] M. Eric and C. Manning, "A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 468–473.
[47] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
[48] D. Blei and J. Lafferty, "Correlated topic models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, Art. no. 147.

Lizi Liao received the PhD degree from the NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore, in 2019. She is currently a research fellow with Next++ Center, School of Computing, National University of Singapore. Her research interests include conversational system and multimedia analysis. Her works have appeared in top-tier conferences such as MM, WWW, ICDE, IJCAI and AAAI, and top-tier journals such as the *IEEE Transactions on Knowledge and Data Engineering*. She received the Best Paper Award Honorable Mention of ACM MM 2018. Moreover, she has served as the PC member for international conferences including SIGIR, WSDM, and the invited reviewer for journals including the *IEEE Transactions on Knowledge and Data Engineering* and the *IEEE Transactions on Multimedia* and the *Knowledge-Based Systems*.

Ryuichi Takanobu received the BE degree from Tsinghua University, Beijing, China, in 2019. He is currently working toward the master's degree in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include conversational system and reinforcement learning. He has published several papers in leading international conferences including WWW, ACL, AAAI, and IJCAI.

Yunshan Ma received the BE degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2015. He is currently working toward the PhD degree in the School of Computing, National University of Singapore, Singapore. His research interests include multimedia information processing, computer vision, and conversational system.

Xun Yang received the PhD degree from the School of Computer and Information, Hefei University of Technology, Hefei, China, in 2018. He is currently a research fellow with NExT++ Center, School of Computing, National University of Singapore. His research interests include information retrieval, computer vision, and multimedia information processing.

Minlie Huang received the PhD degree from Tsinghua University, Beijing, China, in 2006. He is currently an associate professor with the Institute for Artificial Intelligence, Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include deep/reinforcement learning and natural language processing. He has published more than 70 papers in premier conferences and journals (ACL, IJCAI, AAAI, KDD, SIGIR, WWW, etc.). He won IJCAI 2018 Distinguished Paper Award, CCL 2018 Best Demo Award, NLPCC 2015 Best Paper Award, Hanvon Youngth Innovation Award in 2018, and MSRA Collaborative Research Award in 2019. His work, Emotional Chatting Machines, was reported by MIT Technology Review, the Guardian, nVIDIA, and other mass media.

Tat-Seng Chua received the PhD degree from the University of Leeds, Leeds, United Kingdom. He is the KITHCT chair professor with the School of Computing, National University of Singapore. He was the acting and founding dean of the School from 1998–2000. His main research interest include multimedia information retrieval and social media analytics. In particular, his research focuses on the extraction, retrieval and question-answering (QA) of text and rich media arising from the Web and multiple social networks. He is the co-director of Next, a joint Center between NUS and Tsinghua University to develop technologies for live social media search. He is the 2015 winner of the prestigious ACM SIGMM award for Outstanding Technical Contributions to Multimedia Computing, Communications and Applications. He is the chair of steering committee of ACM International Conference on Multimedia Retrieval (ICMR) and Multimedia Modeling (MMM) conference series. He is also the general co-chair of ACM Multimedia 2005, ACM CIVR (now ACM ICMR) 2005, ACM SIGIR 2008, and ACM Web Science 2015. He serves in the editorial boards of four international journals. He is the co-founder of two technology startup companies in Singapore.