Deep Conversational Recommender in Travel

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

Abstract—When traveling to a foreign country, we are often in dire need of an intelligent conversational agent to provide instant and informative responses to our various queries. However, to build such a travel agent is non-trivial. First of all, travel naturally involves several sub-tasks such as hotel reservation, restaurant recommendation and taxi booking *etc*, which invokes the need for global topic control. Secondly, the agent should consider various constraints like price or distance given by the user to recommend an appropriate venue. In this paper, we present a Deep Conversational Recommender (**DCR**) and apply to travel. It augments the sequence-to-sequence (seq2seq) models with a neural latent topic component to better guide response generation and make the training easier. To consider the various constraints for venue recommendation, we leverage a graph convolutional network (GCN) based approach to capture the relationships between different venues and the match between venue and dialog context. For response generation, we combine the topic-based component with the idea of pointer networks, which allows us to effectively incorporate recommendation results. We perform extensive evaluation on a multi-turn task-oriented dialog dataset in travel domain and the results show that our method achieves superior performance as compared to a wide range of baselines.
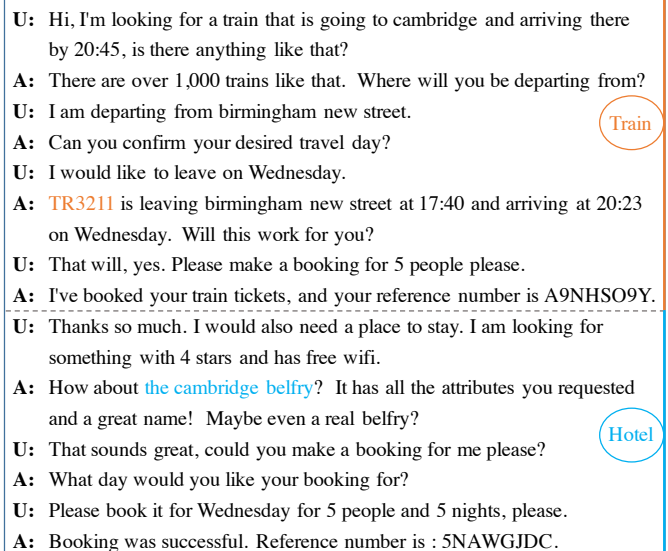
Index Terms—Conversational recommender, Dialog system, Travel Domain.

---◆---

1 INTRODUCTION

Conversational agents and travel go hand in hand. In fact, artificial intelligence is set to be a game-changer for this industry, through helping travellers and companies simplifying travel arrangements and streamlining business procedures. Currently, large companies in travel industry such as Expedia.com, KLM and Booking.com, race to launch their online chatbots. For example, in the first month after KLM lauching their bot service, their volume of Facebook messages jumped 40 percent. It is a fact that conversational agents are revolutionizing customer care, from answering questions to solving customer issues to venue recommendations *etc*.

Although conversational agents in travel show big commercial potential, it is non-trivial to build such an intelligent system to meet the various user needs. As the example illustrated in Figure 1, travel 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. Currently, neural conversational models [1], [2], [3] are the latest development in conversational modeling, where seq2seq-based models, such as HRED [3], 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 dialog sessions. Therefore, current state-of-the-art methods might not be sufficient for the travel agent scenario.

At the same time, to satisfy users' need such as finding hotel or restaurant, the ability to recognize constraints and

---

- L. Liao is the corresponding author. E-mail: liaolizi.llz@gmail.com
- R. Takanobu and M. Huang are with Tsinghua University
- L. Liao, Y. Ma, Y. Xun and TS. Chua are with National University of Singapore.

Fig. 1: A sample dialog between a user (U) and an agent (A) from the dataset. We observe the need for global topic control and appropriate venue recommendation.

efficiently leverage them in venue recommendation is essential for building an intelligent system. For example, in order to generate the hotel 'cambridge belfry' as in Figure 1, the agent needs to capture not only the constraints like '4 stars' and 'free wifi' under the same topic, but also the hidden location constraint 'cambridge'. Moreover, we observe that distance and time are also important factors to consider when doing venue recommendation under the travel scenario. Indeed, there have been some existing efforts aiming to find appropriate venues in conversational systems. Many task-oriented dialog systems try to form queries and feed them to database systems to retrieve venues [4], [5], [6]. However, such methods heavily rely on the exact match of

constraints which is rather sensitive to even slight language variations. It also has other limitations such as the weakness on modeling relationships between venues and the inability to back-propagate error signals from the end output to the raw inputs. To alleviate such problems, memory networks are leveraged to 'softly' incorporate venue entries in external Knowledge Bases (KBs) [7], [8]. However, the various relationships between venues, even the simplest 'nearby' relation, are hard to model. The most recent studies such as [9], [10] integrate conversational system with recommendation components, but the recommendation part only focus on learning the interplay between users and items.

In this paper, we propose a Deep Conversational Recommender (**DCR**) as shown in Figure 2 and apply it to the travel domain to address the above mentioned problems. First, in order to enable the agent to swiftly differentiate subtopics in travel, 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 [11]. Second, we employ a graph convolutional network (GCN) based approach to capture the various relationships between venues and learn the match between venue and dialog context. When generating venue recommendations, the agent ranks the venues by calculating the matching scores between the learned venue representations and dialog context representations. The key idea is that GCN-based component helps the conversational recommender to generate better representations of venues that incorporate both venue feature information as well as venue relations. Third, we combine the topic-based component and the GCN-based component by leveraging the idea of pointer networks. It allows us to effectively incorporate the recommendation results into the response generation procedure.

To sum up, the main contributions of this work are threefold as follows:

- We propose a conversational travel agent 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 jointly captures venue information, relationships between them and the dialog 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, including the neural latent topic component, graph convolutional network based recommender and the response integration mechanism inspired by pointer network. 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

Task-oriented systems aim to assist users to achieve specific goals with natural language such as restaurant reservation and schedule arrangement. Traditionally, they have been built in pipelined fashion: language understanding, dialog management, knowledge query and response generation [12], [13], [14]. However, the requirement of human labor in designing dialog ontology and heavy reliance on slot filling as well as dialog state tracking techniques limits its usage to relatively simple and specific tasks such as flight reservation [15] or querying bus information [16]. For travel which involves multiple sub-tasks and needs to handle various constraints for venues recommendation, such pipelined methods might not be sufficient.

Recently, end-to-end approaches for dialog modeling, which use seq2seq-based models, have shown promising results [3], [5], [17]. They directly map plain text dialog history to the output responses. Since the dialog 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 [18], a stochastic neural dialog 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 [19], [20]. Therefore, progress has been made in conditioning the seq2seq model on coarse-grained knowledge representations, such as a fuzzily-matched retrieval result via attention [21] or a set of pre-organized topic or scenario labels [22], [23]. 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. 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
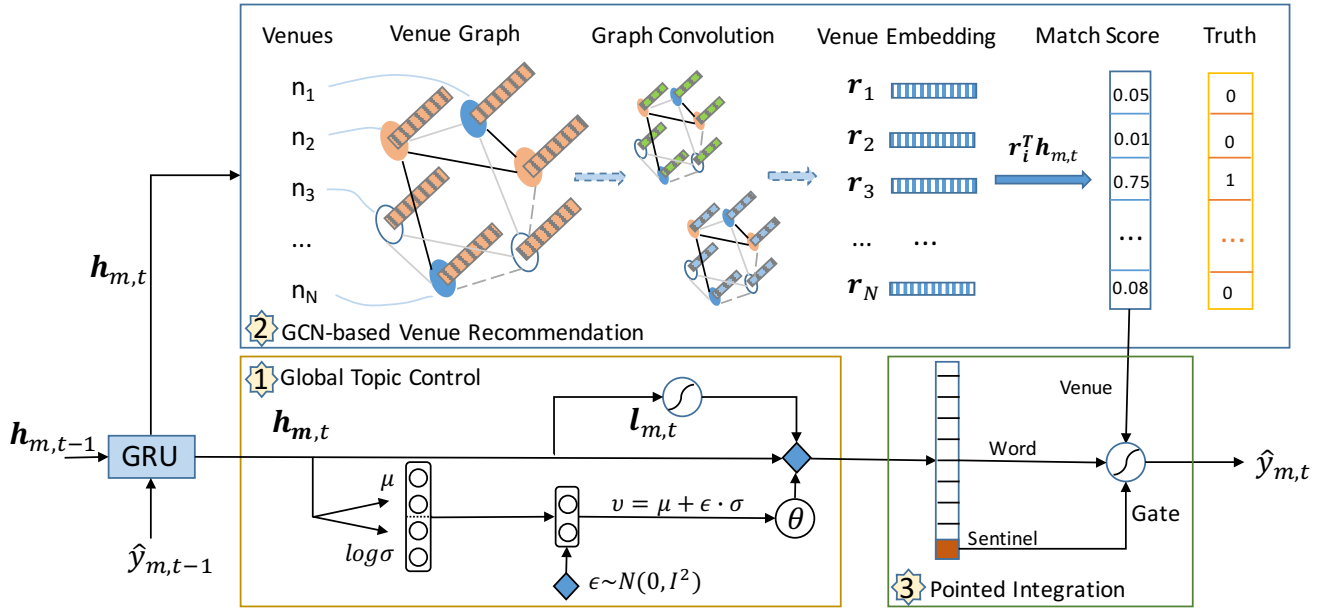
Fig. 2: The proposed DCR model for travel, which consists of three components. The global topic control component enables the system to switch between various sub-tasks quickly. The GCN-based venue recommendation component generates venues by considering venue information, relations among them and the match to dialog context. Finally, a pointed integration mechanism incorporates the two components for the final response generation. The diamonds are stochastic nodes.

[24], [25]. These approaches either rely heavily on tracking the dialog 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, [9] 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 dialog, while ignores other dialog contents expressed in natural language. As another example, [26] leveraged a generative Gaussian model to recommend items to users in a conversation. However, their dialog 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 dialog system often directly solicits facets from users [5], [27]. There are also another line of approaches using reinforcement learning (RL) to train goal-oriented dialog systems [10], [28]. For instance, in [10], a simulated user is used to help train a dialog 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 constraints for venues which are prevalent in travel and seamlessly integrate these results to the response generation.

3   THE DCR MODEL

We aim at building an agent capable of answering users' queries and making venue recommendations to satisfy their requirements. One might therefore characterize our system as a conversational recommender. The complete architecture of our approach is illustrated in Figure 2. Starting from the bottom of Figure 2, there are mainly three sub-components as follows.

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

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

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 dialog contexts $\{u_1, \cdots, u_{m-1}\}$ 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 dialog 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. After formalizing the problem as above, we provide more details for each of these components one by one.

3.1   Global Topic Control

3.1.1   Basic Encoder

Formally, we consider a dialog as a sequence of M utterances $D = \{u_1, \cdots, u_M\}$. Each utterance u_m is a sequence with N_m tokens, i.e. $u_m = \{y_{m,1}, \cdots, y_{m,N_m}\}$. 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 [3] generate a target utterance given a source utterance and dialog history. Given the dialog context $\{u_1, \cdots, u_{m-1}\}$, the goal is to produce a *machine* response u_m that maximizes the conditional probability $u_m^* = argmax_{u_m} p(u_m|u_{m-1}, \cdots, u_1)$. Here, we apply the well-accepted hierarchical recurrent encoder decoder (HRED) model [3] 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 $\mathbf{h}_m$,

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

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

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 state updates that can either be a vanilla RNN cell or complex cell such as LSTM or GRU.

3.1.2 Generative Process

While RNN-based models can theoretically model arbitrarily long dialog histories if provided enough capacity, in practice even the improved version like LSTM or GRU struggles to do so [29], [30]. In dialogs between user and travel agent, there usually exist long-range dependencies captured by sub-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 [11]. 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 the following,

1. Encode the user input u_{m-1} and dialog context C into a vector representation $\mathbf{h}_{m-1} = HRED(u_{m-1}, \cdots, u_1) \in \mathbb{R}^d$.

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 t-th token $y_{m,t}$,

   (a) Update the hidden state $\mathbf{h}_{m,t} = f_{\mathbf{W}_H}(\mathbf{h}_{m,t-1}, \mathbf{y}_{m,t-1})$.

   (b) Draw stop word indicator $l_t \sim Bernoulli(sigmoid(\mathbf{W}^T\mathbf{h}_{m,t}))$.

   (c) 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 [3] which encodes dialog 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 dialog 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

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

Since direct optimization of Equation 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 Equation 5 can therefore be constructed as

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

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 Equation 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.

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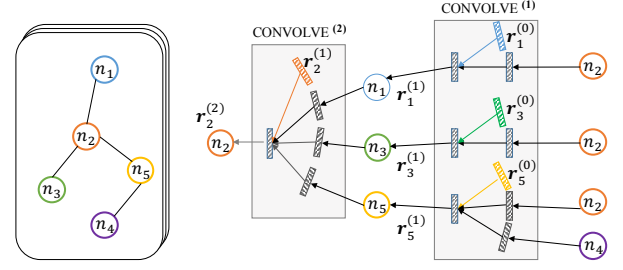\end{aligned}
\tag{8}
$$

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

3.2 GCN-based Venue Recommendation

Given the dialog 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 $O = \{n_1, n_2, \cdots, n_N\}$ is a set of N nodes and $E \subseteq N \times N$ is a set of edges between nodes. Here the nodes can be hotels, restaurants, location area etc, while the relations can be 'nearby' or co-appear etc. In this way, venues located in the same area will be connected closely, and venues co-appeared in the same dialog session will be connected closely. 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 $\widetilde{\mathbf{D}}_{ii} = \sum_j \widetilde{\mathbf{A}}_{ij}$. We denote the attributes of nodes as a matrix $\mathbf{X}$, the representations of nodes in l^{th} layer as $\mathbf{R}^{(l)}$. Initially, we have $\mathbf{R}^{(0)} = \mathbf{X}$ which means that the initial representation of nodes are obtained from embedding node attributes.

Given such a constructed graph, we generate high-quality embeddings or representations of entities that can be used for calculating the matching score with dialog 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 Figure 3, we first project the former layer node

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 $\mathbf{R}^l$.

representation $\mathbf{R}^{(l-1)}$ into a latent space using the weight matrix $\mathbf{W}^{(l)}$ (we omitted the bias term for simplicity),

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

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,

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

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 Equation 9, we present the objective function which encourages the matching between dialog context and venues. Suppose there are M dialog context and ground truth node pairs, we obtain the dialog 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

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

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 dialog context, we can predict the next utterance via the global topic control component and obtain the recommended venue through the GCN-based recommender. To integrate the two lines of results, we propose a pointed integration mechanism. Generally speaking, we use a Gated Recurrent Unit (GRU) [38] to decode the system response. At each decoding step t in turn m, the GRU gets the previously generated token and the previous hidden state as input, and generates the new hidden state,

$$
\mathbf{h}_{m,t} = GRU(\mathbf{h}_{m,t-1}, \hat{\mathbf{y}}_{m,t-1}).
$$

Then the hidden state $\mathbf{h}_{m,t}$ is passed to two branches as shown in Figure 2. In one branch, the $\mathbf{h}_{m,t}$ is passed to the global topic control component. Following the generative process introduced in Subsection 3.1.2, 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 the other branch, the $\mathbf{h}_{m,t}$ is fed to the GCN-based recommender. It helps the recommender rank the venues and output the top ranked venue name.

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

3.3.1 Sentinel

In the final response generation, whether a token is generated from Equation 11 or Equation 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. Otherwise, the model chooses a token in V as the decoded token. Basically, 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 substitute with $. The training objective of this component is $\mathcal{L}_{Topic}$ detailed as Equation 8.

Then we change back the dataset and train the GCN component for venue ranking on it. The dialog context is embedded via the trained global topic control model. The training objective is $\lambda\mathcal{L}_{GCN}$ as detailed in Equation 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 **DCR**, in travel. The experiments are carried out to answer the research questions as follows.

RQ1: Can the proposed DCR properly respond to users' queries in travel? 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?

In what follows, we will first describe the experimental settings. We then answer the above three research questions.

4.1 Experimental Setup

4.1.1 Dataset

Arguably the greatest bottleneck for statistical approaches to dialog system development is the collection of appropriate training dataset, and this is especially true for task-oriented dialog systems [41]. Fortunately, [6] contributed a dataset consisting of over 10K conversation sessions in travel 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 dialog 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 sub-topics — Attraction, Hospital, Police, Hotel, Restaurant, Taxi, Train. The dialogs cover between 1 and 5 sub-topics per dialog thus greatly varying in length and complexity. This broad range of topics captures the common scenarios where sub-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 dialogs collected, where 3, 406 of them focus in single-topic dialogs and 7,032 of them are dialogs 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, around 70% of dialogs 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 dialogs 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** [3]: 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** [6]: It frames the dialog 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 dialogs.

- **Mem2Seq** [8]: It augments the existing MemNN [7] framework with a sequential generative architecture to produce coherent responses for task-oriented dialog systems. It uses global multi-hop attention mechanisms to copy words directly from dialog 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 dialogs.
- **ReDial** [9]: It integrates the HRED based conversational model with a denoising auto-encoder based recommender [42] via a switching mechanism. The recommendation part is pre-trained separately and only considers the co-occurrence of items while ignores the dialog context. The recommender part is also compared in ablation study.
- **NCF** [43]: 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.

4.1.3 Evaluation Protocols

We evaluate the methods in various evaluation protocols. Due to the difficulty in evaluating conversational agents [44], 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 dialogs systems [45]. It is based on the idea of modified n-gram precision, where the higher score denotes better performance.
- **Entity Accuracy**: Similar to [8], we average over the entire set of system responses and compare the entities in plain text. The entities in each gold system response are selected by a predefined entity list. This metric evaluates the ability to recommend appropriate items from the provided item set and to capture the semantics of dialogs [45].

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 and present each of them ten generated dialog 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 dialog session. We allow ties so that multiple methods could be given the same rank for the same

dialog 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 Setups

The proposed model is implemented in PyTorch [1]. 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 pretrained GloVe embeddings [46] 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 [47] with a learning rate of 0.01 is used for training.

We use the Python-based natural language toolkit NLTK to perform tokenization. Entities in dialog sessions are recognized via heuristic rules plus database entries. All counts, time and reference numbers are replaced with the ⟨value_count⟩, ⟨value_time⟩ and ⟨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 ⟨UNK⟩ token. We follow the {S,U,S'} utterance "triples" structure as [3] 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

4.2.1 Corpus-based Evaluation

The result of the corpus-based evaluation is presented in Figure 4 and Figure 5. For each method, the results are obtained based on the best model chosen via the development set. The key observations are as follows.

Overall, the proposed DCR 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% of performance improvement as compared to the second best method, TopicRNN. The two methods perform

---

1. Our code will be made publicly available for reproducibility.

better than all the other baselines. In terms of the entity accuracy score, DCR improves the performance of venue recommendation by 17.2% as compared to the second best method, Mem2Seq. The performance improvements of DCR method demonstrate its effectiveness in travel domain conversational recommendation due to the following aspects: a) DCR 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 dialog contexts which is essential for conversational recommendation.
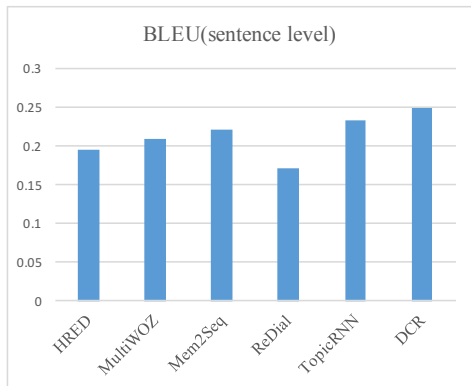

Fig. 4: The BLEU scores for each method.

In more detail, we analyse the BLEU score shown in Figure 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 dialog history and KB — some tokens or entities are directly copied to form responses. It happens frequently that words appeared in dialog 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 dialog context topic and narrow down the generation of tokens. In addition to a similar topic control scheme, D-CR manages to achieve superior performance by achieving better entity prediction.

Regarding the entity accuracy score presented in Figure 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 dialog context. The basic information of venue entities and the relationships between them are ignored. For the MultiWoz method, although a database
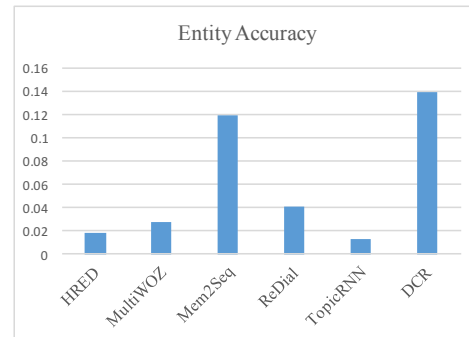

Fig. 5: The entity accuracy scores for each method. Note that this is the top-1 accuracy score since only the top ranked venue is leveraged by text response.

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 dialogs as we detailed before — venue entities appeared in dialog 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 dialog 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 DCR 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 dialog 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 DCR achieves 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 DCR over the other methods are significant. For example, in terms of the Fluency score, DCR improves the performance of response generation by 50.0%, 44.5%, 30.3%, 53.5% and 8.8% 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 DCR might be due to the global topic control mechanism. In travel, dialogs naturally involve multiple sub-tasks, which leads to several topics in the

dialog flow. The topic control component enables the system to swiftly switch among topics and generate within-topic responses.

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
DCR	**3.96**	**3.82**	**1.8**

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 DCR 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 outside knowledge. In DCR, since it already manages to recommend the venue, the slot values are obtained via the venue information through a simple post-process.

For the final ranking of methods, we find it in general accord with the Fluency and Informativeness score trends. The DCR is ranked as the best method by our evaluators, followed by the Mem2Seq method. It actually points out a future direction to enhance our method. Due to the frequent "re-use" phenomenon in dialogs, the dialog context is important to model. To encode it into vector representation as in HRED is not sufficient, direct incorporation of the tokens as in the Mem2Seq method opens a new auxiliary road.
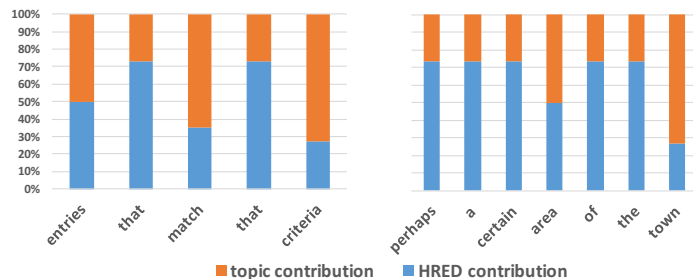
4.3 Analysis on Components

In this subsection, we explore the performance and contribution of the major components in our design. We first evaluate the global topic control component. We showcase the learned topic words, the stop word indicator prediction results and topic distribution of some example dialogs. Then, we explore the recommendation component. We compare the GCN-based recommender with several state-of-the-art recommendation methods and provide the results.

TABLE 2: Four 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

4.3.1 Topic Control of Dialogues

Here we evaluate the performance of the global topic control component. At first, we show whether the learned topic words are coherent. We run the component on our 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.

**Fig. 6: Analysis of the learned stop word indicators.**

Since stop words largely affect the performance of our global topic control component, we also show the analysis of the stop word indicator l. Some examples are shown in Figure 6. As can be seen, the learned indicators correspond to the human intuition, and help to coordinate the contribution of the global topic part and the local syntactic part while generating responses.

To show whether the global topic control component correctly captures the overall topic distribution of dialog sessions, we plot the topic distribution θ of dialogs as shown in Figure 7. Two example dialog session 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 sub-tasks involved during the dialog 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 *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 dialog. Thus, two topics are picked up by the model. We observe that topic 10 corresponds to *hotel* and topic 19 corresponds to *train*.

4.3.2 Venue Recommendation Analysis

In this subsection, we analysis our GCN-based venue recommendation component in detail. The common user-item interaction situation is abstracted from the dialogs by treating the dialog 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

U: Are there any cheap Indian restaurants available?
A: Definitely! I have 3 in the center and 1 in the north. I recommend Kohinoor, at 74 Mill Road City Centre.
U: Can you book a table for me there? Wednesday at 18:30, please, and I'll be dining solo.
A: Great! Your booking for Kohinoor was successful. The reference number is XJBRHRDC. Is there anything else I can do to help?
U: Yes, I am also looking for somewhere to go in the centre of town.
A: What type of attraction do you want?
U: What would you suggest?
A: I would suggest visiting one of the famous colleges.
U: I need a college in the centre of town.
A: We have thirteen to chose from! All but three are free. Would you like me to list them for you?
U: Which one would you suggest and I need the postcode.
A: I would recommend Christ's College. It's free to visit.
U: Could I have their postcode please?
A: Christ's College 's post code is cb23bu.
U: Thank you, that is all the information I need.
A: You are welcome, please contact us if we can help you with anything else.
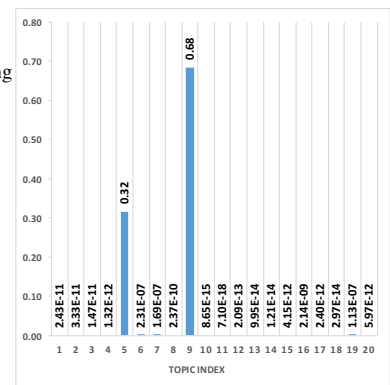
U: I am looking for a hotel, Carolina Bed and Breakfast.
A: Sure, I have it right here. Its a moderate priced guesthouse with a 4 star rating, free internet and parking. Phone number is 01223247015 can I book it for you?
U: No, that is all for now.
A: Ok. I can also help with restaurant, train or taxi reservations. Do you need any of these?
U: I am needing a train to arrive in cambridge by 14:45.
A: What day did you need a train for? Also, where will you be departing from?
U: I want to travel on Tuesday. I'm coming from Broxbourne to Cambridge.
A: TR8199 matches your criteria. It leaves at 13:32 on tuesday. Would you like to book this train?
U: As long as it arrives by 14:45, then it's good. Can you book it for 7 people?
A: It arrives at 14:32. Your reference number is R6MSYW4P and the total fee for 7 people is 125.29 GBP. Is there anything else I can help you with?
U: No, that's everything. Thank you.
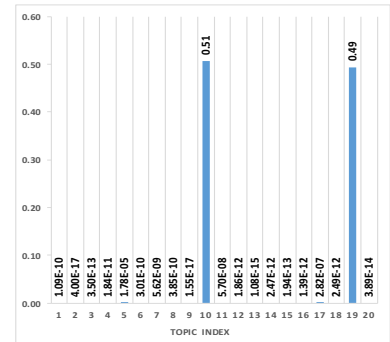**A:** You are very welcome!

Fig. 7: Inferred topic distribution of two example dialog sessions. It shows that some of the topics have been picked up depending on the dialog content.

in the same dialog session are extracted to form the entity vector.

TABLE 3: Performance comparison of recommenders.

Methods	ReDial	NCF	GCN-based
Top-1 Accuracy	0.1065	0.1882	**0.2420**

The results are shown in Table 3. Since often only the top item is leveraged in the dialogs, we report the Top-1 accuracy here. It shows that the GCN-based recommender component achieves better performance as compared to ReDial and NCF methods. For the ReDial recommender, it projects the entity appearance vector v of each dialog 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 dialog context information are largely ignored. At the same time, the entity co-occurrence matrix formed via training dialog sessions is rather sparse. These factors together lead to its relatively weak performance. Regarding the NCF method, the dialog 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 DCR manages to handle all the three evidence sources — the venue information, relations between them and the match to dialog context.

5 CONCLUSION

In order to build an intelligent conversational agent in travel domain, 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 dialog context topics, which narrows down the generation of tokens in decoding. At the same time, a graph convolutional network based recommender manages to pop venues by modeling the venue information, relations between them and the match to dialog context. 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 conversational dataset in travel. Experimental results showed that the proposed DCR method outperformed a wide range of baselines and demonstrated the effectiveness of it in generating fluent and informative responses.

In future, we will continue our work in two directions. First, we will explore the "re-use" phenomenon to further boost the performance of response generation. Second, we will try to leverage extra venue adoption data from travel e-commerce sites to enhance the recommendation performance.

ACKNOWLEDGMENTS

REFERENCES

[1] O. Vinyals and Q. Le, "A neural conversational model," *arXiv preprint arXiv:1506.05869*, 2015.

[2] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015, pp. 1577–1586.

[3] 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 *Thirty AAAI Conference on Artificial Intelligence*, 2016, pp. 3776–3784.

[4] A. Bordes and J. Weston, "Learning end-to-end goal-oriented dialog," in *The 3nd International Conference on Learning Representations*, 2016, pp. 1–14.

[5] T. Wen, D. Vandyke, N. Mrkšíc, M. Gašíc, L. Rojas-Barahona, P. Su, S. Ultes, and S. Young, "A network-based end-to-end trainable task-oriented dialogue system," in *15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017, pp. 438–449.

[6] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gasic, "Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 5016–5026.

[7] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," in *Advances in neural information processing systems*, 2015, pp. 2440–2448.

[8] A. Madotto, C.-S. Wu, and P. Fung, "Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 1468–1478.

[9] R. Li, S. E. Kahou, H. Schulz, V. Michalski, L. Charlin, and C. Pal, "Towards deep conversational recommendations," in *Advances in Neural Information Processing Systems*, 2018, pp. 9748–9758.

[10] Y. Sun and Y. Zhang, "Conversational recommender system," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 235–244.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, pp. 993–1022, 2003.

[12] J. D. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech & Language*, pp. 393–422, 2007.

[13] C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura, "Statistical dialog management applied to wfst-based dialog systems," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 2009, pp. 4793–4796.

[14] S. Young, M. Gašíc, B. Thomson, and J. D. Williams, "Pomdp-based statistical spoken dialog systems: A review," *Proceedings of the IEEE*, pp. 1160–1179, 2013.

[15] S. Seneff and J. Polifroni, "Dialogue management in the mercury flight reservation system," in *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems*, 2000, pp. 11–16.

[16] A. Raux, B. Langner, D. Bohus, A. W. Black, and M. Eskenazi, "Let's go public! taking a spoken dialog system to the real world," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[17] T. Zhao, A. Lu, K. Lee, and M. Eskenazi, "Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 2017, pp. 27–36.

[18] T.-H. Wen, Y. Miao, P. Blunsom, and S. Young, "Latent intention dialogue models," in *International Conference on Machine Learning*, 2017, pp. 3732–3741.

[19] C. Matuszek, J. Cabral, M. J. Witbrock, and J. DeOliveira, "An introduction to the syntax and content of cyc." in *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, 2006, pp. 44–49.

[20] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1400–1409.

[21] M. Ghazvininejad, C. Brockett, M.-W. Chang, B. Dolan, J. Gao, W.-t. Yih, and M. Galley, "A knowledge-grounded neural conversation model," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 5110–5117.

[22] D. Wang, N. Jojic, C. Brockett, and E. Nyberg, "Steering output style and topic in neural response generation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2140–2150.

[23] C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, and W.-Y. Ma, "Topic augmented neural response generation with a joint attention mechanism. arxiv preprint," *arXiv preprint arXiv:1606.08340*, 2016.

[24] C. A. Thompson, M. H. Goker, and P. Langley, "A personalized system for conversational recommendations," *Journal of Artificial Intelligence Research*, pp. 393–428, 2004.

[25] C. Greco, A. Suglia, P. Basile, and G. Semeraro, "Converse-et-impera: Exploiting deep learning and hierarchical reinforcement learning for conversational recommender systems," in *Conference of the Italian Association for Artificial Intelligence*, 2017, pp. 372–386.

[26] K. Christakopoulou, F. Radlinski, and K. Hofmann, "Towards conversational recommender systems," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 815–824.

[27] B. Dhingra, L. Li, X. Li, J. Gao, Y.-N. Chen, F. Ahmed, and L. Deng, "Towards end-to-end reinforcement learning of dialogue agents for information access," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 484–495.

[28] X. Li, Y.-N. Chen, L. Li, J. Gao, and A. Celikyilmaz, "End-to-end task-completion neural dialogue systems," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, 2017, pp. 733–743.

[29] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, pp. 157–166, 1994.

[30] A. B. Dieng, C. Wang, J. Gao, and J. Paisley, "Topicrnn: A recurrent neural network with long-range semantic dependency," *arXiv preprint arXiv:1611.01702*, 2016.

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

[32] A. Mnih and K. Gregor, "Neural variational inference and learning in belief networks," in *International Conference on Machine Learning*, 2014, pp. 1791–1799.

[33] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *International Conference on Machine Learning*, 2016, pp. 1727–1736.

[34] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *The 2nd International Conference on Learning 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 *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 974–983.

[36] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[37] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2016, pp. 1–14.

[38] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning*, 2014.

[39] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," in *Proceedings of the 54th Annual Meeting of the Association for Computational 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 *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 1073–1083.

[41] L. Liao, Y. Ma, X. He, R. Hong, and T.-s. Chua, "Knowledge-aware multimodal dialogue systems," in *2018 ACM Multimedia Conference on Multimedia Conference*, 2018, pp. 801–809.

[42] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 111–112.

[43] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 173–182.

[44] 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."

[45] M. Eric and C. Manning, "A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017, pp. 468–473.

[46] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing*, 2014, pp. 1532–1543.

[47] D. P. Kingma and B. Jimmy, "Adam: A method for stochastic optimization," in *The 3nd International Conference on Learning Representations*, 2015, pp. 1–14.

[48] D. Blei and J. Lafferty, "Correlated topic models," *Advances in neural information processing systems*, p. 147, 2006.