Discovering Dialog Structure Graph for Coherent Dialog Generation

Jun Xu[1]*, Zeyang Lei[2]*, Haifeng Wang[2], Zheng-Yu Niu[2], Hua Wu[2], Wanxiang Che[1]†
[1]Research Center for Social Computing and Information Retrieval,
Harbin Institute of Technology, Harbin, China
[2]Baidu Inc., Beijing, China
{jxu, car}@ir.hit.edu.cn, {leizeyang, wanghaifeng, niuzhengyu, wu_hua}@baidu.com

Abstract

Learning discrete dialog structure graph from human-human dialogs yields basic insights into the structure of conversation, and also provides background knowledge to facilitate dialog generation. However, this problem is less studied in open-domain dialogue. In this paper, we conduct **unsupervised discovery of discrete dialog structure** from chitchat corpora, and then leverage it to facilitate coherent dialog generation in downstream systems. To this end, we present an unsupervised model, Discrete Variational Auto-Encoder with Graph Neural Network (DVAE-GNN), to discover discrete hierarchical latent dialog states (at the level of both session and utterance) and their transitions from corpus as a dialog structure graph. Then we leverage it as background knowledge to facilitate dialog management in a RL based dialog system. Experimental results on two benchmark corpora confirm that DVAE-GNN can discover meaningful dialog structure graph, and the use of dialog structure as background knowledge can significantly improve multi-turn coherence.

1 Introduction

With the aim of building a machine to converse with humans naturally, some work investigate neural generative models (Shang et al., 2015; Serban et al., 2017). While these models can generate locally relevant dialogs, they struggle to organize individual utterances into globally coherent flow (Yu et al., 2016; Xu et al., 2020b). The possible reason is that it is difficult to control the overall dialog flow without background knowledge about dialog structure.[1] However, due to the complexity of open-domain conversation, it is laborious and costly to annotate dialog structure manually. Therefore, it is of great importance to discover open-domain dialog structure from corpus in an unsupervised way for coherent dialog generation.

Some studies tried to discover dialog structure from task-oriented dialogs (Shi et al., 2019). However, the number of their dialog states is limited to only dozens or hundreds, which cannot cover fine-grained semantics in open-domain dialogs. Furthermore, the dialog structures they discovered generally only contain utterance-level semantics (non-hierarchical), without session-level semantics (chatting topics) that are essential in open-domain dialogs (Wu et al., 2019; Kang et al., 2019; Xu et al., 2020c).[2] Thus, in order to provide a full picture of open-domain dialog structure, it is desirable to discover a two-layer directed graph that contains session-level semantics in the upper-layer vertices, utterance-level semantics in the lower-layer vertices, and edges among these vertices.

In this paper, we propose a novel discrete variational auto-encoder with graph neural network (**DVAE-GNN**) to discover a two-layer dialog structure from chitchat corpus. Intuitively, since discrete dialog states are easier to capture transitions for dialog coherence, we use discrete variables to represent dialog states (or vertices in the graph) rather than dense continuous ones in most VAE-based dialog models (Serban et al., 2017; Zhao et al., 2017). Specifically, we employ an RNN Encoder with softmax function as vertex recognition module in DVAE, and an RNN decoder as reconstruction module in DVAE, as shown in Figure 3. Furthermore, we integrate GNN into DVAE to model complex relations among discrete variables for more effective discovery. The parameters of DVAE-GNN can be optimized by minimizing a reconstruction loss, without the requirement of any annotated datasets.

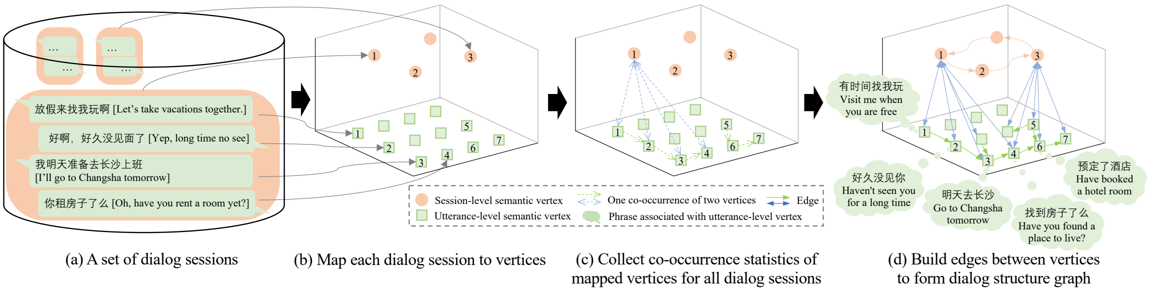
[*] Equal contribution.

[†] Corresponding author: Wanxiang Che.

[1] Dialog structure means dialog states and their transitions.

[2] A session refers to a dialog fragment about one topic.

*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing*, pages 1726–1739
August 1–6, 2021. ©2021 Association for Computational Linguistics


…
…
…
…
放假来找我玩啊 [Let's take vacations together.]
好啊，好久没见面了 [Yep, long time no see]
我明天准备去长沙上班
[I'll go to Changsha tomorrow]
你租房子了么 [Oh, have you rent a room yet?]
1
2
3
4
5
6
7
Session-level semantic vertex
Utterance-level semantic vertex
One co-occurrence of two vertices
Edge
Phrase associated with utterance-level vertex
有时间找我玩
Visit me when you are free
好久没见你
Haven't seen you for a long time
明天去长沙
Go to Changsha tomorrow
找到房子了么
Have you found a place to live?
预定了酒店
Have booked a hotel room

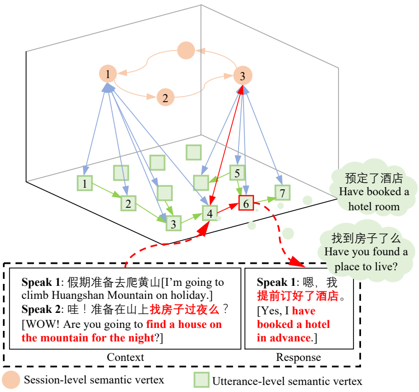
(a) A set of dialog sessions

(b) Map each dialog session to vertices

(c) Collect co-occurrence statistics of mapped vertices for all dialog sessions

(d) Build edges between vertices to form dialog structure graph

Figure 1: The procedure of dialog structure discovery. Figure (d) shows the discovered dialog structure graph.


Speak 1: 假期准备去爬黄山[I'm going to climb Huangshan Mountain on holiday.]
Speak 2: 哇！准备在山上**找房子过夜么**？[WOW! Are you going to **find a house on the mountain for the night**?]

Speak 1: 嗯，我**提前订好了酒店**。[Yes, I **have booked a hotel in advance**.]

1
2
3
4
5
6
7
预定了酒店
Have booked a hotel room
找到房子了么
Have you found a place to live?
Context
Response
Session-level semantic vertex
Utterance-level semantic vertex

Figure 2: Response generation grounded on a dialog structure graph.

As shown in Figure 1, with well-trained DVAE-GNN, we build the dialog structure graph by three steps. **First**, we map all dialog sessions to utterance-level and session-level vertices, as shown in Figure 1 (b); **Second**, we calculate co-occurrence statistics of mapped vertices for all dialog sessions, as shown in Figure 1 (c).[3] **Finally**, we build edges among vertices based on all collected co-occurrence statistics to form the dialog structure graph, as shown in Figure 1 (d).

To prove the effectiveness of the discovered structure, we propose a hierarchical reinforcement learning (RL) based graph grounded conversational system (**GCS**) to leverage it for conversation generation. As shown in Figure 2, given a dialog context, GCS first maps it to a utterance-level vertex, and then learns to walk over graph edges, and finally selects a contextual appropriate utterance-level vertex to guide response generation at each turn.

Our contribution includes: (1) we identify the task of unsupervised dialog structure graph discovery in open-domain dialogs. (2) we propose a novel model, DVAE-GNN, for hierarchical dialog structure graph discovery. Experimental results on two benchmark corpora demonstrate that we can discover meaningful dialog structure, the use of GNN is crucial to dialog structure discovery, and the graph can improve dialog coherence significantly.

2 Related Work

2.1 Dialog structure learning for task-oriented dialogs

There are previous work on discovering human-readable dialog structure for task-oriented dialogs via hidden Markov models (Chotimongkol, 2008; Ritter et al., 2010; Zhai and Williams, 2014) or variational auto-encoder (Shi et al., 2019). However, the number of their dialog states is limited to only dozens or hundreds, which cannot cover fine-grained semantics in chitchat. Moreover, our method can discover a hierarchical dialog structure, which is different from the non-hierarchical dialog structures in most previous work.

2.2 Knowledge aware conversation generation

There are growing interests in leveraging knowledge bases for generation of more informative responses (Moghe et al., 2018; Dinan et al., 2019; Liu et al., 2019; Xu et al., 2020c,a). In this work, we employ a dialog-modeling oriented graph built from dialog corpora, instead of a external knowledge base, in order to facilitate multi-turn dialog modeling.

2.3 Latent variable models for chitchat

Recently, latent variables are utilized to improve diversity (Serban et al., 2017; Zhao et al., 2017; Gu et al., 2019; Gao et al., 2019; Ghandeharioun et al., 2019), control responding styles (Zhao et al., 2018; Li et al., 2020) and incorporate knowledge (Kim et al., 2020) in dialogs. Our work differs from

[3] Co-occurrence means that two utterance-level vertices are mapped by two adjacent utterances in a session.

[4] ai.baidu.com/tech/nlp_basic/dependency_parsing

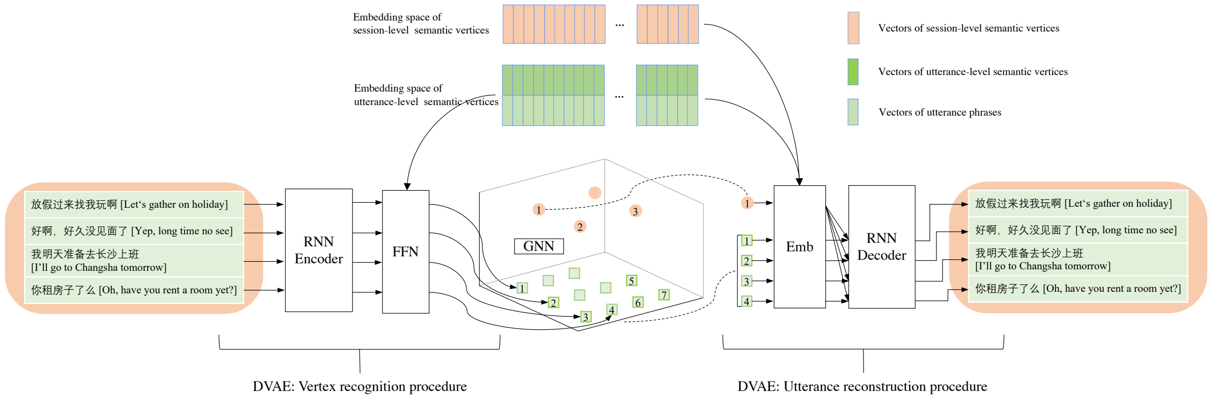

Embedding space of
session-level semantic vertices
Embedding space of
utterance-level semantic vertices
...
Vectors of session-level semantic vertices
Vectors of utterance-level semantic vertices
Vectors of utterance phrases
放假过来找我玩啊 [Let's gather on holiday]
好啊，好久没见面了 [Yep, long time no see]
我明天准备去长沙上班
[I'll go to Changsha tomorrow]
你租房子了么 [Oh, have you rent a room yet?]
RNN
Encoder
FFN
GNN
1
2
3
4
5
6
7
Emb
RNN
Decoder
DVAE: Vertex recognition procedure
DVAE: Utterance reconstruction procedure

Figure 3: Overview of our algorithm "DVAE-GNN" for discovering a dialog structure graph from dialog dataset. FFN denotes feed-forword neural networks and Emb refers to embedding layers

Algorithm 1 Phrase extraction

```
Input: An utterance $U$
Output: A set of phrases $E$ extracted from $U$
Obtain a dependency parse tree $T$ for $U$;[4]
Get all the head words HED that are connected to ROOT
  node, and all the leaf nodes in $T$ (denoted as $L$);
for each leaf node in $|L|$ do
    Extract a phrase consisting of words along the tree
      from HED to current leaf node, denoted as $e_i$;
    If $e_i$ is a verb phrase, then append it into $E$;
end for
return $E$
```

theirs in that: (1) we focus on open-domain dialog structure discovery. (2) we use discrete latent variables to model dialog states instead of dense continuous ones in most previous work.

3 Our Approach

3.1 Problem Definition

Given a corpus D that contains $|D|$ dialog sessions $\{X_1, X_2, ..., X_{|D|}\}$, where each dialog session X consists of a sequence of c utterances, and $X = [x_1, ..., x_c]$. The objective is to discover a two-layer dialog structure graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ from **all dialog sessions** in D, where $\mathcal{V}$ is the vertex set and $\mathcal{E}$ is the edge set. Specifically, $\mathcal{V}$ consists of two types, v^s_m ($1 \leq m \leq M$) for session-level vertices (topics) and v^u_n ($1 \leq n \leq N$) for utterance-level vertices. $\mathcal{E}$ contains three types: edges between two session-level vertices (denoted as Sess-Sess edges), edges between two utterance-level vertices (denoted as Utter-Utter edges), and edges between an utterance-level vertex and its parent session-level vertices (denoted as Sess-Utter edges).

Figure 3 shows the proposed DVAE-GNN framework. It contains two procedures, vertex recognition that maps utterances and sessions to vertices (as the role of recognition module in VAE (Kingma and Welling, 2014)), and utterance reconstruction that regenerates all utterances in sessions (as the role of decoding module in VAE).

3.2 Graph Initialization

Vertex Initialization. Theoretically, we can cold start the representation learning of vertices in dialog structure graph. In practice, to accelerate the learning procedure, we warm start each utterance-level vertex representation with the combination of two parts: one discrete latent variable and one distinct phrase. The associated phrase with each utterance-level vertex provides prior semantic knowledge for utterance-level vertex representation, which is beneficial for reducing the learning difficulty. Specifically, we first extract distinct phrases from all dialog utterances with Algorithm 1. Then we choose the top-N most frequent extracted phrases (the same number as utterance-level vertices), and then randomly match utterance-level vertices and the phrases in pairs during initialization. Notice that the association relations are not changed afterwards.

Formally, we use $\mathbf{\Lambda}_s$ and $\mathbf{\Lambda}_x$ to represent the hidden representation matrix of discrete session-level and utterance-level vertices respectively. The calculation can be shown as follows:

$$\mathbf{\Lambda}_s[m] = W^s \boldsymbol{v}^s_m \tag{1}$$

$$\mathbf{\Lambda}_x[n] = [\boldsymbol{e}(ph_n); W^u \boldsymbol{v}^u_n], \tag{2}$$

where $\mathbf{\Lambda}_s[m]$ denotes the representation vector of m-th session-level vertex, $\mathbf{\Lambda}_x[n]$ denotes the representation vector of n-th utterance-level vertex, $\boldsymbol{v}^s_m$ and $\boldsymbol{v}^u_n$ are one-hot vectors of discrete vertices, $\boldsymbol{e}(ph_n)$ denotes the representation vector of the associated phrase ph_n with $\boldsymbol{v}_n^u$, W^u and W^s are parameters, and ";" denotes concatenation operation. Specifically, for phrase representation, we first feed word sequence in the phrase to an RNN encoder and obtain their hidden vectors. Then we compute the average pooling value of these hidden vectors as $\boldsymbol{e}(ph_n)$.

Edge Initialization We build an initial Utter-Utter edge between two utterance-level vertices when their associated phrases can be extracted sequentially from two adjacent utterances in the same dialog session.

3.3 Vertex Recognition

Utterance-level Vertex Recognition. For each utterance x_i in a dialog session, we map it to an utterance-level vertex. Specifically, we first encode the utterance x_i with an RNN encoder to obtain its representation vector $\boldsymbol{e}(x_i)$. Then, we calculate the posterior distribution of the mapped utterance-level vertex, z_i, by a feed-forward neural network (FFN):

$$z_i \sim q(z|x_i) = \text{softmax}(\boldsymbol{\Lambda}_x \boldsymbol{e}(x_i)). \quad (3)$$

Finally, we obtain the mapped utterance-level vertex, z_i, by sampling from the posterior distribution with Gumbel-Softmax (Jang et al., 2017). Here, we can obtain an utterance-level vertex sequence after mapping each utterance in one dialog session, where the sequence is utilized for session-level vertex recognition.

Session-level Vertex Recognition. We assume that each session-level vertex corresponds to a group of similar utterance-level vertex sequences that are mapped by different dialog sessions. And these similar sequences might have overlapped utterance-level vertices. To leverage this locally overlapping vertex information for encouraging mapping similar utterance-level vertex sequences to similar session-level vertices, we employ graph neural network to model complex relations among vertices for session-level vertex recognition. Specifically, we utilize a three-layer graph convolution network (GCN) over Utter-Utter edges to calculate structure-aware utterance-level semantics. The calculation is defined by:

$$\boldsymbol{h}^j_{v^u_n} = \sigma^j\Big(\sum_{v^u_{n'} \in \mathcal{N}(v^u_n)} \boldsymbol{h}^{j-1}_{v^u_{n'}}\Big), \quad (4)$$

where $\boldsymbol{h}^j_{v^u_n}$ denotes the j-th layer structure-aware representation for the n-th utterance-level vertex v^u_n. σ^j is the *sigmoid* activation function for the j-th layer, and $\mathcal{N}(v^u_n)$ is the set of utterance-level neighbors of v^u_n in the graph. Here, we can obtain a structure-aware semantic sequence $[\boldsymbol{h}^3_{v^u_{z_1}}, \boldsymbol{h}^3_{v^u_{z_i}}, ..., \boldsymbol{h}^3_{v^u_{z_c}}]$, where $\boldsymbol{h}^3_{v^u_{z_i}}$ represents the final structure-aware representation of i-th mapped utterance-level vertex, $v^u_{z_i}$.

Then, we feed the structure-aware semantic sequence to an RNN encoder, denoted as the vertex-sequence encoder, to obtain the structure-aware session representation $\boldsymbol{e}(z_{1,...,c})$. We calculate the posterior distribution of the mapped session-level vertex, g, as follows:

$$g \sim q(g|z_{1,...,c}) = \text{softmax}(\boldsymbol{\Lambda}_s \boldsymbol{e}(z_{1,...,c})). \quad (5)$$

Then, we obtain the mapped session-level vertex, g, by sampling from the session-level posterior distribution with Gumbel-Softmax.

3.4 Utterance Reconstruction

We reconstruct all utterances in the dialog session by feeding these mapped vertices into an RNN decoder (denoted as the reconstruction decoder). Specifically, to regenerate utterance x_i, we concatenate the representation vector of mapped utterance-level vertex $\boldsymbol{\Lambda}_x[z_i]$ and the representation vector of the mapped session-level vertex $\boldsymbol{\Lambda}_s[g]$, as the initial hidden state of the reconstruction decoder.

Finally, we optimize the DVAE-GNN model by maximizing the variational lower-bound (ELBO) (Kingma and Welling, 2014). Please refer to Appendix D for more details.

3.5 Graph Construction

After training DVAE-GNN, we construct the dialog structure graph with well-trained DVAE-GNN by three steps, as shown in Figure 1. Specifically, we first map all dialog sessions in corpus to vertices by Equation 3 and 5.

Then, we collect co-occurrence statistics of these mapped vertices. Specifically, we count the total mapped times for each session-level vertex, denoted as $\#(v^s_i)$, and those for each utterance-level vertex, denoted as $\#(v^u_j)$. Furthermore, we collect the co-occurrence frequency of a session-level vertex and an utterance-level vertex that are mapped by a dialog session and an utterance in it respectively, denoted as $\#(v^s_i, v^u_j)$. Moreover, we collect the co-occurrence frequency of two utterance-level vertices that are sequentially mapped by two adjacent utterances in a dialog session, denoted as $\#(v^u_j, v^u_k)$.

Finally, we build edges between vertices based on these co-occurrence statistics. We first build a directed Utter-Utter edge from v_j^u to v_k^u if the bi-gram transition probability $\#(v_j^u, v_k^u)/\#(v_j^u)$ is above a threshold α^{uu}. Then, we build a bidirectional Sess-Utter edge between v_j^u and v_k^s if the probability $\#(v_i^s, v_j^u)/\#(v_j^u)$ is above a threshold α^{su}. Moreover, we build a directed Sess-Sess edge from v_i^s to v_o^s, if $\#(v_i^s, v_o^s)/\#(v_i^s)$ is above a threshold α^{ss}, where the first item $\#(v_i^s, v_o^s)$ is the number of utterance-level vertices that are connected to both session-level vertices. Here, Sess-Sess edges are dependent on Sess-Utter edges.

3.6 Graph Grounded Dialog Generation

To prove the effectiveness of the discovered structure for coherent dialog generation, we utilize a graph grounded conversation system (GCS) following (Xu et al., 2020a). Different from single-layer policy in Xu *et al.*(Xu et al., 2020a), we present a hierarchical policy for two-level vertex selection. The GCS contains three modules: (1) a dialog context understanding module that maps given dialog context (the previous two utterances) to an utterance-level vertex (called as hit utterance-level vertex) in the graph with well-trained DVAE-GNN, (2) a hierarchical policy that learns to walk over one-hop graph edges (for dialog coherence) to select an utterance-level vertex to serve as response content, and (3) a response generator that generate an appropriate response based on the selected utterance-level vertex. Specifically, a session-level sub-policy first selects a session-level vertex as current dialog topic. Then, an utterance-level sub-policy selects an utterance-level vertex from current dialog topic's child utterance-level vertices.

Session-level sub-policy Let $\mathcal{A}_{s_l}^g$ denote the set of session-level candidate actions at time step l. It consists of all parent session-level vertices of the hit utterance-level vertex. Given current RL state s_l at the time step l, the session-level sub-policy μ^g selects an appropriate session-level vertex from $\mathcal{A}_{s_l}^g$ as the current dialog topic. Specifically, μ^g is formalized as follows:

$$\mu^g(s_l, v_{c_j^g}^s) = \frac{\exp(\boldsymbol{e}_{s_l}{}^T \boldsymbol{\Lambda}_s[c_j^g])}{\sum_{k=1}^{N_l^g} \exp(\boldsymbol{e}_{s_l}{}^T \boldsymbol{\Lambda}_s[c_k^g])},$$

where $\boldsymbol{e}_{s_l}$ is the aforementioned RL state representation, c_j^g the j-th session-level vertex in $\mathcal{A}_{s_l}^g$, and N_l^g is the number of session-level vertices in $\mathcal{A}_{s_l}^g$.

Utterance-level sub-policy Let $\mathcal{A}_{s_l}^u$ denote the set of utterance-level candidate actions at time step l. It consists of utterance-level vertices that are connected to the vertex of current dialog topic. Given current state s_l at the time step l, the utterance-level sub-policy μ^u selects an optimal utterance-level vertex from $\mathcal{A}_{s_l}^u$. Specifically, μ^u is defined as follows:

$$\mu^u(s_l, v_{c_j^u}^u) = \frac{\exp(\boldsymbol{e}_{s_l}{}^T \boldsymbol{\Lambda}_x[c_j^u])}{\sum_{k=1}^{N_l^u} \exp(\boldsymbol{e}_{s_l}{}^T \boldsymbol{\Lambda}_x[c_k^u])}.$$

Here, $\boldsymbol{e}_{s_l}$ is the aforementioned RL state representation, c_j^u is the j-th utterance-level vertex in $\mathcal{A}_{s_l}^u$, and N_l^u is the number of utterance-level candidate vertices in $\mathcal{A}_{s_l}^u$. With the distribution calculated by the above equation, we utilize Gumbel-Softmax to sample an utterance-level vertex from $\mathcal{A}_{s_l}^u$, to provide response content for response generator, which is a Seq2Seq model with attention mechanism.

To train RL, we use a set of rewards including utterance relevance, utter-topic closeness, and repetition penalty. For the session-level sub-policy, its reward r^g is the average rewards from the utterance-level sub-policy during current dialog topic. The reward for the utterance-level sub-policy, r^u, is a weighted sum of the below-mentioned factors. The default values of weights are set as [60, 0.5, -0.5]. [5]

i) Utterance relevance We choose the classical multi-turn response selection model, DAM in (Zhou et al., 2018), to calculate utterance relevance. We expect the generated response is coherent to dialog context.

ii) Utter-topic closeness The selected utterance-level vertex v_j^u should be closely related to current topic v_i^s. And we use the $\#(v_i^s, v_j^u)/\#(v_j^u)$ in Section 3.5 as the utter-topic closeness score.

iii) Repetition penalty This factor is 1 when the selected utterance-level vertex shares more than 60% words with one of contextual utterance, otherwise 0. We expect that the selected utterance-level vertices are not only coherent, but also diverse.

Further implementation details can be found in the Appendix C.

4 Experiments for Dialog Structure Graph Discovery

4.1 Datasets and Baselines

We evaluate the quality of dialog structure graph discovered by our method and baselines on two

[5] We optimize these weights by grid search.

Datasets	Methods	Automatic Evaluation		Human Evaluation		
		NLL	BLEU-1/2.	S-U Appr. (Multi-turn coherence)	U-U Appr. (Single-turn coherence)	Sess.V.-Qual. (Multi-turn coherence)
Weibo	DVRNN	29.187	0.427/0.322	-	0.16	-
	Phrase Graph	-	-/-	-	0.63	-
	DVAE-GNN	**20.969**	**0.588/0.455**	**0.85**	**0.79**	**1.44**
	DVAE-GNN w/o GNN	23.364	0.560/0.429	0.53	0.78	1.06
	DVAE-GNN w/o phrase	24.282	0.468/0.355	0.43	0.27	0.95
Douban	DVRNN	72.744	0.124/0.093	-	0.14	-
	Phrase Graph	-	-/-	-	0.34	-
	DVAE-GNN	**35.975**	**0.525/0.412**	**0.60**	**0.70**	**0.93**
	DVAE-GNN w/o GNN	37.415	0.504/0.394	0.38	0.54	0.48
	DVAE-GNN w/o phrase	49.606	0.254/0.206	0.28	0.19	0.27

Table 1: Evaluation results for dialog structure graphs extracted from Weibo corpus or Douban corpus. As DVRNN learns only utterance-level states, its results in terms of S-U Appr. and Sess.V.-Qual. are not available.

benchmark datasets: (1) **Weibo** (Li and Yan, 2018): this is a Chinese multi-turn tweet-style corpora. After data cleaning, we obtain 3.1 million sessions for training, 10k sessions for validation and 10k sessions for testing. (2) **Douban** (Wu et al., 2017): we use the original multi-turn dialog corpus, and obtain 2.3 million sessions for training, 10k sessions for validation and 10k sessions for testing. For the Weibo or Douban corpus, each dialog session has 4 sentences on average, and each sentence contains about 7 or 14 words respectively. The discovered dialog structure graph on Weibo corpus contains 1,641,238 utterance-level vertices, 6000 session-level vertices and 11,561,007 edges. And the discovered dialog structure graph on Douban corpus contains 1,768,720 utterance-level vertices, 5500 session-level vertices and 6,117,159 edges. The number of utterance-level vertices is equal to the number of extracted phrase number in corpus and session-level vertices is determined by grid search based on the NLL metric in Section 4.2.

In this work, we select DVRNN (Shi et al., 2019) as a baseline, since there is few previous study on unsupervised open-domain dialog structure discovery. DVRNN is the SOTA unsupervised method in discovering dialog structure in task-oriented dialogs, which outperforms other hidden Markov based methods by a large margin (Shi et al., 2019). We rerun the original source codes.[6] Notice that, to suite the setting of open-domain dialog and also consider the limit of our 16G GPU memory (we set batch size as 32 to ensure training efficiency), we set the number of dialog states as 50 (originally it is 10).[7] We also evaluate the quality of the initialized graph (denoted as Phrase Graph) that consists of only phrases (as vertices) and initial edges (between phrases) in Section 3.2. For more details, please refer to Appendix A.1.

4.2 Evaluation Metrics

We evaluate discovered dialog structure graph with both automatic evaluation and human evaluation. For automatic evaluation, we use two metrics to evaluate the performance of reconstruction: (1) **NLL** is the negative log likelihood of dialog utterances; (2) **BLEU-1/2** measures how much that reconstructed sentences contains 1/2-gram overlaps with input sentences (Papineni et al., 2002). The two metrics indicate how well the learned dialog structure graph can capture important semantic information in dialog dataset.

Further, we manually evaluate the quality of edges and vertices in the graph. For edges, (1) **S-U Appr.** for multi-turn dialog coherence. It measures the appropriateness of Sess-Utter edges, where these edges provide crucial prior information to ensure multi-turn dialog coherence (see results in Section 5.4). "1" if an utterance-level vertex is relevant to its session-level vertex (topic), otherwise "0". (2) **U-U Appr.** for single-turn dialog coherence: It measures the quality of Utter-Utter edges between two utterance-level vertices, where these edges provide crucial prior information to

[6] github.com/wyshi/Unsupervised-Structure-Learning

[7] We ever tried to modify their codes to support the learning of a large number of dialog states (up to 30k). But its performance is even worse than original code with 50 states.

ensure single-turn dialog coherence. It is "1" if an Utter-Utter edge is suitable for responding, otherwise "0". Notice that we don't evaluate the quality of Sess-Sess edges because Sess-Sess edges are dependent on the statistics of Sess-Utter edges.

Meanwhile, for vertices, we evaluate **Session-level Vertex Quality (Sess.V.-Qual.)**. Ideally, a session-level vertex (topic) should be mapped by dialog sessions that share high similarity. In other words, we can measure the quality of a session-level vertex by evaluating the similarity of semantics between two sessions that are mapped to it. It is "2" if the two sessions mapped to the same session-level vertex are about the same or highly similar topic, "0" if the two sessions contains different topic, otherwise "1". Specifically, during evaluation, we provide typical words of each topic by calculating TF-IDF on utterances that are mapped to it. High "Sess.V.-Qual." is beneficial to conduct topic management for coherent multi-turn dialogs. Note that we don't evaluate utterance-level vertex quality since it is too fine-grained for annotators to determine whether two utterances that are mapped to a utterance-level vertex are "highly-similar".

For human evaluation, we sample 300 cases and invite three annotators from a crowd-sourcing platform to evaluate each case.[8] Notice that all system identifiers are masked during human evaluation.

4.3 Experiment Results

As shown in Table 1, DVAE-GNN significantly outperforms DVRNN, in terms of all the metrics (sign test, p-value < 0.01) on the two datasets. It demonstrates that DVAE-GNN can better discover meaningful dialog structure graph. Specifically, DVAE-GNN obtains the best results in terms of NLL and BLEU-1/2, which shows that DVAE-GNN can better capture important semantic information in comparison with DVRNN. Meanwhile, DVAE-GNN also surpasses all baselines in terms of "U-U Appr." and "S-U Appr.". It indicates that our discovered dialog structure graph has higher-quality edges and can better facilitate coherent dialog generation.

Furthermore, we conduct ablation study. Specifically, to evaluate the contribution of GNN, we remove GNN from DVAE-GNN, denoted as DVAE-GNN w/o GNN. We see that its performance drop sharply in terms of "S-U Appr." and "Sess.V.-Qual.". It demonstrates that GNN can better incorporate the structure information (complex relations among vertices) into session-level vertex representation learning. Moreover, to evaluate the contribution of phrases to utterance-level vertex representation, we remove phrases, denoted as DVAE-GNN w/o phrase. We see that its scores in terms of all the metrics drops sharply, especially the three human evaluation metrics. The reason is that it's difficult to learn high-quality utterance-level vertex representation from a large amount of fine-grained semantic content in open-domain dialogs without any prior information. The Kappa value is above 0.4, showing moderate agreement among annotators.

Two sample parts of the discovered dialog structure graph can be found in Appendix B.

5 Experiments for Graph Grounded Dialog Generation

To confirm the benefits of discovered dialog structure graph for coherent conversation generation, we conduct experiments on the graph discovered from Weibo corpus. All the systems (including baselines) are trained on Weibo corpus.

5.1 Models

We carefully select the following six baselines.

MMPMS It is the multi-mapping based neural open-domain conversational model with posterior mapping selection mechanism (Chen et al., 2019), which is a SOTA model on the Weibo Corpus.

MemGM It is the memory-augmented open-domain dialog model (Tian et al., 2019), which learns to cluster U-R pairs for response generation.

HRED It is the hierarchical recurrent encoder-decoder model (Serban et al., 2016).

CVAE It is the Conditional Variational Auto-Encoder based neural open-domain conversational model (Zhao et al., 2017).

VHCR-EI This variational hierarchical RNN model can learn hierarchical latent variables from open-domain dialogs (Ghandeharioun et al., 2019). It is a SOTA dialog model with hierarchical VAE.

DVRNN-RL It discovers dialog structure graph for task-oriented dialog modeling (Shi et al., 2019).

GCS It is our proposed dialog structure graph grounded dialog system with hierarchical RL.

GCS w/ UtterG It is a simplified version of GCS that just uses the utterance-level graph and utterance-level sub-policy.

GCS w/ Phrase Graph It is a simplified version of GCS that just uses the phrase graph and utterance-level sub-policy.

[8] test.baidu.com

Methods	Coherence		Informativeness		Overall Quality	
	Multi.T.-Coh.*	Single.T.-Coh.*	Info.*	Dist-1/2#	Enga.*	Length#
MMPMS	0.66	0.45	0.50	0.08/0.32	0.24	5.82
MemGM	0.53	0.37	0.34	0.09/0.33	0.20	4.08
HRED	0.54	0.43	0.19	0.08/0.26	0.20	5.04
CVAE	0.58	0.39	0.43	0.11/0.38	0.22	7.74
VHCR-EI	0.68	0.43	0.53	0.12/0.36	0.28	7.30
DVRNN-RL	0.60	0.39	0.39	0.06/0.22	0.22	7.86
GCS	**1.03**	**0.59**	**0.58**	**0.19/0.55**	**0.48**	**8.00**
GCS w/ UtterG	0.93	0.56	0.55	0.16/0.47	0.34	**8.00**
GCS w/ Phrase Graph	0.72	0.41	0.54	0.16/0.45	0.24	**8.00**

Table 2: Evaluation results for baselines and our system trained on Weibo corpus. $*$ or $\#$ denote human or automatic evaluation metrics.

We use the same user simulator for RL training of DVRNN-RL, GCS and GCS w/ UtterG. Here, we use the original MMPMS as user simulator because it achieves the best result on the Weibo Corpus. The user simulator is pre-trained on dialog corpus and not updated during policy training. We use the original source codes for all the baselines and the simulator. Further details about baselines and GCS can be found in Appendix A.2.

We conduct model-human dialogs for evaluation. Given a model, we first randomly select an utterance (the first utterance in a session) from test set for the model side to start the conversations with a human turker. Then the human is asked to converse with the selected model till 8 turns are reached. Finally, we obtain 50 model-human dialogs for multi-turn evaluation. Then we randomly sample 200 U-R pairs from the above dialogs for single-turn evaluation.

5.2 Evaluation Metrics

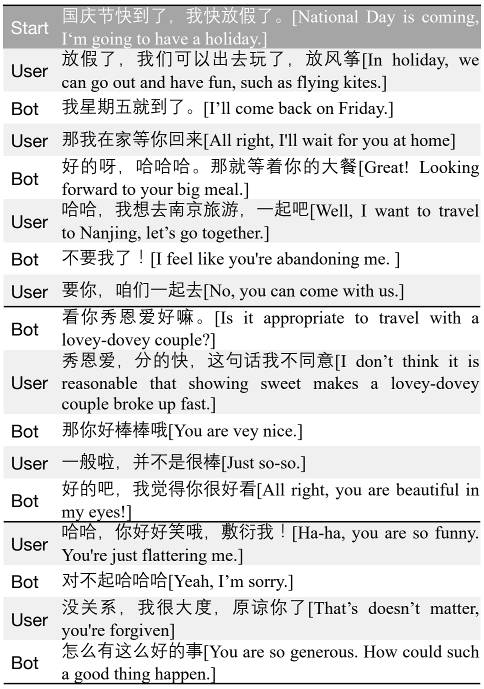
Since the proposed system does not aim at predicting the highest-probability response at each turn, but rather the long-term success of a dialog (e.g., coherence), we do not employ BLEU (Papineni et al., 2002) or perplexity for evaluation. We use three multi-turn evaluation metrics and three single-turn metrics. For human evaluation, we invite three annotators to conduct evaluation on each case, and we ask them to provide 1/0 (Yes or No) scores for most of the metrics. Moreover, for multi-turn coherence, we first ask the annotators to manually segment a dialog by topics and then conduct evaluation on each session. A session refers to a dialog fragment about one topic. Notice that system identifiers are masked during human evaluation.

Multi-turn Metrics. We use the following metrics: (1) **Multi-turn Coherence (Multi.T.-Coh.)** It measures the coherence within a session. Common incoherence errors in a session include amphora errors across utterances and information inconsistency. “0” means that there are more than two incoherence errors in a session. “1” means that there are only one error. “2” means that there are no errors. Finally, we compute the average score of all the sessions. (2) **Dialog engagement (Enga.)** This metric measures how interesting a dialogs is. It is “1” if a dialog is interesting and the human is willing to continue the conversation, otherwise “0”. (3) **Length of high-quality dialog (Length)** A high-quality dialog ends if the model tends to produce dull responses or two consecutive utterances are highly overlapping (Li et al., 2016b).

Single-turn Metrics. We use the following metrics: (1) **Single-turn Coherence (Single.T.-Coh.)** “0” if a response is inappropriate as an reply, otherwise “1”; (2) **Informativeness (Info.)** “0” if a response is a “safe” response, e.g. “I don’t know”, or it is highly overlapped with context, otherwise “1”; (3) **Distinct (Dist.-i)** It is an automatic metric for response diversity (Li et al., 2016a).

5.3 Experiment Results

As shown in Table 2, GCS significantly outperforms all the baselines in terms of all the metrics except “Length-of-dialog” (sign test, p-value < 0.01). It indicates that GCS can generate more coherent, informative and engaging dialogs. Specifically, our system’s two sub-policies strategy on the dialog structure graph enables more coherent dialog flow control than hierarchical latent variable based VHCR-EI model that performs the best among



Start	国庆节快到了，我快放假了。[National Day is coming, I‘m going to have a holiday.]
User	放假了，我们可以出去玩了，放风筝[In holiday, we can go out and have fun, such as flying kites.]
Bot	我星期五就到了。[I’ll come back on Friday.]
User	那我在家等你回来[All right, I'll wait for you at home]
Bot	好的呀，哈哈哈。那就等着你的大餐[Great! Looking forward to your big meal.]
User	哈哈，我想去南京旅游，一起吧[Well, I want to travel to Nanjing, let’s go together.]
Bot	不要我了！[I feel like you're abandoning me.]
User	要你，咱们一起去[No, you can come with us.]
Bot	看你秀恩爱好嘛。[Is it appropriate to travel with a lovey-dovey couple?]
User	秀恩爱，分的快，这句话我不同意[I don’t think it is reasonable that showing sweet makes a lovey-dovey couple broke up fast.]
Bot	那你好棒棒哦[You are vey nice.]
User	一般啦，并不是很棒[Just so-so.]
Bot	好的吧，我觉得你很好看[All right, you are beautiful in my eyes!]
User	哈哈，你好好笑哦，敷衍我！[Ha-ha, you are so funny. You're just flattering me.]
Bot	对不起哈哈哈[Yeah, I’m sorry.]
User	没关系，我很大度，原谅你了[That’s doesn’t matter, you're forgiven]
Bot	怎么有这么好的事[You are so generous. How could such a good thing happen.]

Figure 4: A sample dialog between our dialog system GCS and a human, where“Bot” is our system and “User” is the human. This dialog contains three dialog topics. We translate the original Chinese texts into English language.

baselines, as indicated by “Multi.T.-Coh.”. Moreover, our high-quality edges between utterance-level vertices (measured by the metric “U-U Appr.” in Table 1) help GCS to achieve higher single-turn coherence score than DVRNN-RL, as indicated by “Single.T.-Coh.”. In addition, GCS, VHCR-EI, MMPMS and CVAE can obtain better performance in terms of “Info.”, indicating that latent variable can effectively improve response informativeness. The Kappa value is above 0.4, showing moderate agreement among annotators.

5.4 Case Study of Conversation Generation

Figure 4 shows a sample dialog between our system “GCS” and a human. We see that our system can generate a coherent, engaging and informative multi-turn dialog. For an in-depth analysis, we manually segment the whole dialog into two sessions. It can be seen that the first session is about “meeting appointment”, and it contains a reasonable dialog logic, I will have a holiday → I will arrive → wait for you at home → look forward to a big meal. And the second session is about “joking between friends”, and it also contains a reasonable logic, you are beautiful → flattering me → I am sorry.

Ablation Study. In order to evaluate the contribution of session-level vertices, we run GCS with an utterance-level dialog structure graph, denoted as “GCS w/ UtterG”. Results in Table 2 show that its performance in terms of “Multi.T.-Coh.” and “Enga.” drops sharply. It demonstrates the contribution of our hierarchical dialog structure graph for enhancing dialog coherence and dialog engagement. The possible reason for the inferior performance of “GCS w/ UtterG” is that the removal of session-level vertices harms the capability of selecting coherent utterance-level vertex sequence.

6 Conclusion

In this paper, we conduct unsupervised discovery of discrete dialog structure from chitchat corpora. Further, we try to formalize the structure as a two-layer directed graph. To discover the dialog structure, we present an unsupervised model, DVAE-GNN, which integrates GNN into DVAE to model complex relations among dialog states for more effective dialog structure discovery. Experimental results demonstrate that DVAE-GNN can discover meaningful dialog structure, and the use of dialog structure as background knowledge can significantly improve multi-turn dialog coherence.

Acknowledgments

We are grateful for the support from Ying Yu. This work is supported by the National Key Research and Development Project of China (No.2018AAA0101900) and the National Natural Science Foundation of China (NSFC) via grant 61976072.

References

Chaotao Chen, Jinhua Peng, Fan Wang, Jun Xu, and Hua Wu. 2019. Generating multiple diverse responses with multi-mapping and posterior mapping selection. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4918–4924. ijcai.org.

Ananlada Chotimongkol. 2008. Learning the structure of task-oriented conversations from the corpus of in-domain dialogs. In *Ph.D. thesis, Carnegie Mellon University*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li, Guodong Zhou, and Shuming Shi. 2019. A discrete CVAE for response generation on short-text conversation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1898–1908, Hong Kong, China. Association for Computational Linguistics.

Asma Ghandeharioun, Judy Hanwen Shen, Natasha Jaques, Craig Ferguson, Noah Jones, Àgata Lapedriza, and Rosalind W. Picard. 2019. Approximating interactive human evaluation with self-play for open-domain dialog systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13658–13669.

Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. 2019. Dialogwae: Multimodal response generation with conditional wasserstein autoencoder. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y-Lan Boureau, and Jason Weston. 2019. Recommendation as a communication game: Self-supervised bot-play for goal-oriented dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1951–1961, Hong Kong, China. Association for Computational Linguistics.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential latent knowledge selection for knowledge-grounded dialogue. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Juntao Li and Rui Yan. 2018. Overview of the nlpcc 2018 shared task: Multi-turn human-computer conversations. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 446–451. Springer.

Zhibin Liu, Zheng-Yu Niu, Hua Wu, and Haifeng Wang. 2019. Knowledge aware conversation generation with explainable reasoning over augmented graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1782–1792, Hong Kong, China. Association for Computational Linguistics.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of Twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180, Los Angeles, California. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3776–3784. AAAI Press.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3295–3301. AAAI Press.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. 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 (Volume 1: Long Papers)*, pages 1577–1586, Beijing, China. Association for Computational Linguistics.

Weiyan Shi, Tiancheng Zhao, and Zhou Yu. 2019. Unsupervised dialog structure learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1797–1807, Minneapolis, Minnesota. Association for Computational Linguistics.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT Press.

Zhiliang Tian, Wei Bi, Xiaopeng Li, and Nevin L. Zhang. 2019. Learning to abstract for memory-augmented conversational response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3816–3825, Florence, Italy. Association for Computational Linguistics.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804, Florence, Italy. Association for Computational Linguistics.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 496–505, Vancouver, Canada. Association for Computational Linguistics.

Jun Xu, Zeyang Lei, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. 2020a. Enhancing dialog coherence with event graph grounded content planning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3941–3947. ijcai.org.

Jun Xu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020b. Conversational graph grounded policy learning for open-domain conversation generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1835–1845, Online. Association for Computational Linguistics.

Jun Xu, Haifeng Wang, Zhengyu Niu, Hua Wu, and Wanxiang Che. 2020c. Knowledge graph grounded goal planning for open-domain conversation generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9338–9345.

Zhou Yu, Ziyu Xu, Alan W Black, and Alexander Rudnicky. 2016. Strategy and policy learning for non-task-oriented conversational systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 404–412, Los Angeles. Association for Computational Linguistics.

Ke Zhai and Jason D. Williams. 2014. Discovering latent structure in task-oriented dialogues. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–46, Baltimore, Maryland. Association for Computational Linguistics.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. 2018. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1098–1107, Melbourne, Australia. Association for Computational Linguistics.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, Vancouver, Canada. Association for Computational Linguistics.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1127, Melbourne, Australia. Association for Computational Linguistics.

A Implementation Details

A.1 Implementation Details about DVAE-GNN

For all models, we share the same vocabulary (maximum size is 50000) and initialized word embedding (dimension is 200) with the pre-trained Tencent AI Lab Embedding.[9] Meanwhile, we randomly initialized the embedding space of session-level vertices and latent vectors for utterance-level vertices (dimensions are 200). The hidden sizes of all RNN encoders and RNN decoders are set as 512. The three threshold variables about co-occurrence statistics α^{uu}, α^{su} and α^{ss} are all set as 0.05.

We use the PaddlePaddle framework for the development of our systems.[10]

Notice that it is costly to calculate Equation 3 in Section 3.3 since the total number of utterance-level vertices, N, is very large (more than one million). In practice, for each utterance, we first retrieve the top-50 most related utterance-level vertices according to Okapi BM25 (Robertson and Zaragoza, 2009) similarity between the utterance and associated phrases of all candidate vertices. And then calculate Equation 3 only with these retrieved vertices. Thus, only a part of vectors in $\mathbf{\Lambda}_x$ will be dynamically updated for each training sample when training DVAE-GNN.

A.2 Experiment settings about GCS

Source codes about baselines

- HRED:github.com/julianser/hed-dlg-truncated
- MMPMS:github.com/PaddlePaddle/Research/tree/master/NLP/IJCAI2019-MMPMS
- CVAE:github.com/snakeztc/NeuralDialog-CVAE
- VHCR-EI:github.com/natashamjaques/neural_chat
- MemGM:github.com/tianzhiliang/MemoryAugDialog
- DVRNN:github.com/wyshi/Unsupervised-Structure-Learning

Hyper-parameter Setting for Training In our experiments, all the models share the same vocabulary (maximum size is 50000 for both Weibo corpus and Douban corpus), initialized word embedding (dimension is 200) with the Tencent AI Lab Embedding. Moreover, bidirectional one-layer GRU-RNN (hidden size is 512) is utilized for all the RNN encoders and RNN decoders. In addition, dropout rate is 0.3, batch size is 32 and optimizer is Adam(lr=2le-3) for all models. During RL training, the discounting weight for rewards is set as 0.95. The MMPMS model for the user simulator employs 10 responding mechanisms. We utilize dependency parse for phrase extraction.[11] We pre-train the response generator in the Weibo Corpus.

Rewards and Training Procedure for the Graph grounded Conversational System. We use the PaddlePaddle framework for the development of our systems.[12] We hot-start the response generator by pre-training it before the training of policy module. Meanwhile, to make the RL based training process more stable, we employ the A2C method (Sutton and Barto, 2018) for model optimization rather than the original policy gradient as done in previous work (Li et al., 2016b). Moreover, during RL training, the parameters of the policy module are updated, and the parameters of response generator and the representation of semantic vertices stay intact.

B Case Study of Dialog Structure Graph Discovery

Figure 5 shows a part of the unified dialog structure graph that is discovered from the Weibo corpus. Each yellow-colored circle in this figure represents a session-level vertex with expert interpreted meanings based on the information of top words (from phrases of utterance-level vertices belonging to this session-level vertex) ranked by TF/IDF. Each green-colored rectangle represents an utterance-level vertex. The directed-arrows between utterance-level vertices represent dialog transitions between states, and the utterance-level vertices within blue dotted-lines are about the same session-level vertex (topic).

We observe reasonable dialog structures in Figure 5. It captures the major interaction logic in dialogs about the topic "go traveling", traveling is really good → you decide where to travel → let's

[9] ai.tencent.com/ailab/nlp/embedding.html

[10] paddlepaddle.org.cn/

[11] ai.baidu.com/tech/nlp_basic/dependency_parsing

[12] paddlepaddle.org.cn/

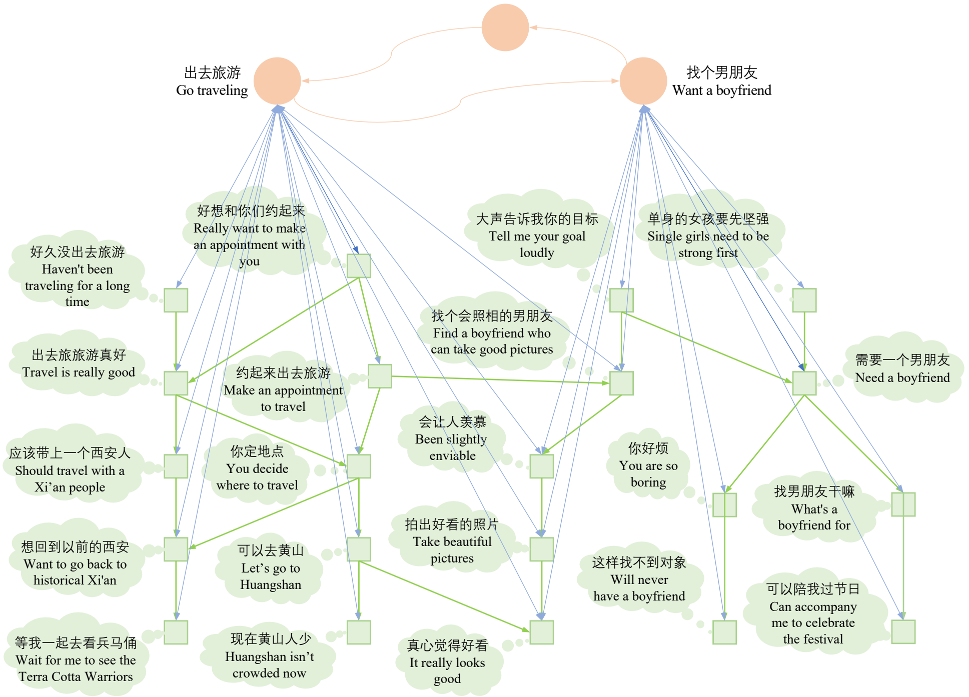


Figure 5: A part of the unified dialog structure graph that is extracted from Weibo corpus. Here, we interpret session-level semantics based on their child utterance-level vertices. We translate the original Chinese texts into English language.

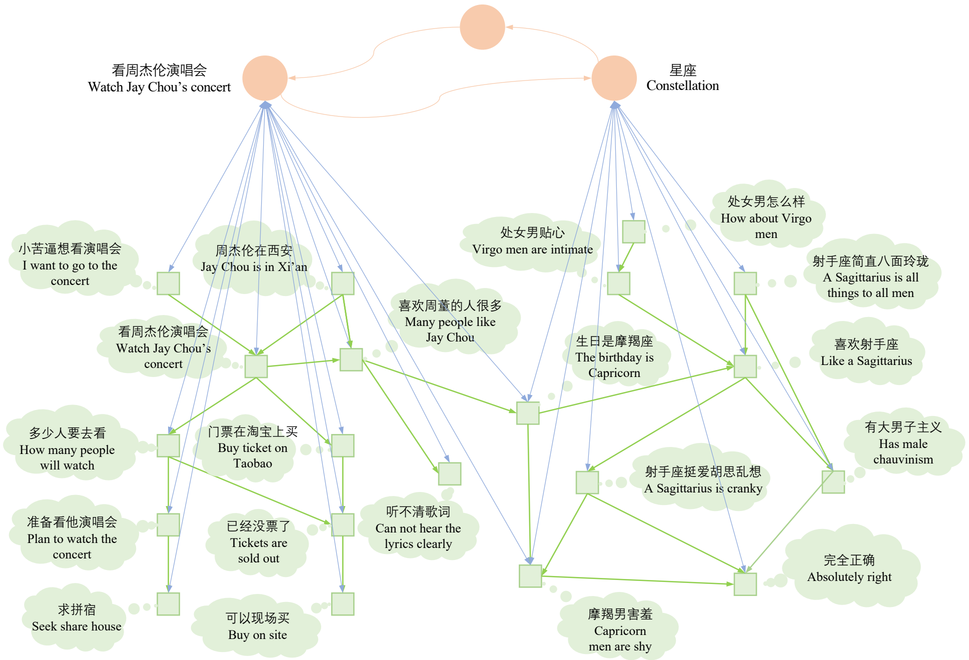


Figure 6: A part of the unified dialog structure graph that is extracted from Weibo corpus. Here, we interpret session-level semantics based on their child utterance-level vertices. We translate the original Chinese texts into English language.

go to Huangshan → comments about Huangshan. Furthermore, it also captures the major logic in dialogs about the topic "want a boyfriend", need a boyfriend → why? → he can accompany me to celebrate the festival. Moreover, it captures a dialog topic transition between the topic "go trveling" and another topic "want a boyfriend".

Figure 6 shows another part of the unified dialog structure graph that discovered from the Weibo corpus.

C GCS with RL

In the following, we will elaborate the details of GCS.

C.1 Dialog Context Understanding

Given a dialog context (the last two utterances), we first map it to the graph by recognizing the most related utterance-level vertex with the well-trained DVAE-GNN. Here, the recognized utterance-level vertex is denoted as the hit utterance-level vertex.

For policy learning, we build current RL state s_l at time step l by collecting dialog context (the last two utterances), previously selected session-level vertex sequence, and previously selected utterance-level vertex sequence. Here, we first utilize three independent RNN encoders to encode them respectively, and then concatenate these three obtained representation vectors, to obtain the representation of the RL state, $\boldsymbol{e}_{s_l}$.

C.2 Response Generator

The response generator is a pre-trained Seq2Seq model with attention mechanism, whose parameters are not updated during RL training. Specifically, we take the last user utterance, and the associated phrase of the selected utterance-level vertex as input of the generator.

D Training Objective for DVAE-GNN

The proposed DVAE-GNN model consists of two procedures. In the recognition procedure, for a dialog session X that consists of a sequence of c utterances, $X = [x_1, ..., x_c]$, in recognition procedure, we first recognize an utterance-level vertex z_i for each utterance x_i, and then recognize a session-level vertex g based on all recognized utterance-level vertices, $[z_1, ..., z_c]$. In reconstruction procedure, we regenerate all the utterances in X with the predicted vertices $Z = [z_1, ..., z_c, g]$. Here, we optimize the proposed DVAE-GNN model by maximizing the variational lower-bound:

$$\mathbb{E}_{q(Z|X)}[\log p(X|Z)] - KL(q(Z|X)\|p(Z)),$$

where $p(Z)$ is the prior uniform distribution of Z.

Specifically, we approximate the first item in the above equation by sampling Z from $q(Z|X)$ and calculate the the negative log-likelihood reconstruction loss. For the second item, we calculate it by:

$$\sum_{j=1}^{c} KL[q(z_j|x_j)\|p(z_j)] + KL[q(g|z_{1,...,c})\|p(g)],$$

where we can calculate each sub-item straightly since $z_{1,...,c}$ and g follow discrete distribution. Below, we provide the derivation of the second item.

$$\begin{aligned}
&KL[q(Z|X)\|p(Z)] \\
&= \sum_{Z}[\log q(Z|X) - \log p(Z)]q(Z|X) \\
&= \sum_{z_{1,...,c},g} \{\sum_{j=1}^{c}[\log q(z_j|x_j) - \log p(z_j)] + [\log q(g|z_{1,...,c}) - \\
&\quad \log p(g)]\} \prod_{i=1}^{c} q(z_i|x_i)q(g|z_{1,...,c}) \\
&= \sum_{j=1}^{c} \sum_{z_{1,...,c},g} [\log q(z_j|x_j) - \log p(z_j)]q(z_j|x_j) \prod_{i=1,i\neq j}^{c} q(z_i|x_i)q(g|z_{1,...,c}) \\
&\quad + \sum_{z_{1,...,c},g} [\log q(g|z_{1,...,c}) - \log p(g)]q(g|z_{1,...,c}) \prod_{i=1}^{c} q(z_i|x_i) \\
&= \sum_{j=1}^{c} \sum_{z_j}[\log q(z_j|x_j) - \log p(z_j)] \sum_{z_{[1,...,c]-j},g} q(z_j|x_j) \prod_{i=1,i\neq j}^{c} q(z_i|x_i)q(g|z_{1,...,c}) \\
&\quad + \sum_{g}[\log q(g|z_{1,...,c}) - \log p(g)]q(g|z_{1,...,c}) \sum_{z_{1,...,c}} \prod_{i=1}^{c} q(z_i|x_i) \\
&= \sum_{j=1}^{c} KL[q(z_j|x_j)\|p(z_j)] \sum_{z_{[1,...,c]-j},g} \prod_{i=1,i\neq j}^{c} q(z_i|x_i)q(g|z_{1,...,c}) \\
&\quad + KL[q(g|z_{1,...,c})\|p(g)] \sum_{z_{1,...,c}} \prod_{i=1}^{c} q(z_i|x_i) \\
&= \sum_{j=1}^{c} KL[q(z_j|x_j)\|p(z_j)] + KL[q(g|z_{1,...,c})\|p(g)]
\end{aligned}$$