Task-Oriented Clustering for Dialogues

[1,2]Chenxu Lv, [1]Hengtong Lu, [2]Shuyu Lei, [2]Huixing Jiang,
[2]Wei wu, [1]Caixia Yuan, [1]Xiaojie Wang
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan Group, Beijing, China
{chenxulv, luhengtong, yuancx, xjwang}@bupt.edu.cn
{leishuyu, jianghuixing, wuwei30}@meituan.com

Abstract

A reliable clustering algorithm for task-oriented dialogues can help developer analysis and define dialogue tasks efficiently. It is challenging to directly apply prior normal text clustering algorithms for task-oriented dialogues, due to the inherent differences between them, such as coreference, omission and diversity expression. In this paper, we propose a Dialogue Task Clustering Network (DTCN) model for task-oriented clustering. The proposed model combines context-aware utterance representations and cross-dialogue utterance cluster representations for task-oriented dialogues clustering. An iterative end-to-end training strategy is utilized for dialogue clustering and representation learning jointly. Experiments on three public datasets show that our model significantly outperformed strong baselines in all metrics[1].

1 Introduction

Task-Oriented Dialogue Clustering (TODC) aims to group task-oriented dialogues into different clusters according to their underlying tasks. Since each cluster includes dialogues for one specific task, it therefore brings convenience for task induction and definition. Especially for large unlabeled human-human dialogues, TODC can be employed to help to induce and define new tasks rapidly which is important for designing of task-oriented dialogue system.

Most prior studies focus on normal text clustering, and have made significant progress via keywords extracting (Bafna et al., 2016; Neto et al., 2000), topic model (Blei et al., 2001; Onan et al., 2017), deep clustering (Xie et al., 2016; Guo et al., 2017; Jiang et al., 2017; Yang et al., 2017). However, inherent differences between task-oriented dialogues and normal texts make above methods difficult to be applied in clustering of task-oriented dialogues directly. The first difficulty is that coreference and information omission occur frequently in dialogues (Su et al., 2019), which makes it harder to build a good representation for utterances in dialogue than in normal text. The second difficulty is that the task-related slot names and intents are scattered in each utterance implicitly and expressed diversely. In most cases, only slots values are given in dialogues without explicit slot names. Only by comparing utterances in different dialogues, we can find task-related implicit information, as shown in Fig.1. Considering these special characteristics in task-oriented dialogues, we emphasize that TODC should utilize in-dialogue relations between different utterance to build context-aware representations for each utterance and utilize cross-dialogue similarity between utterances in different dialogues to induce implicit task-related concepts information.

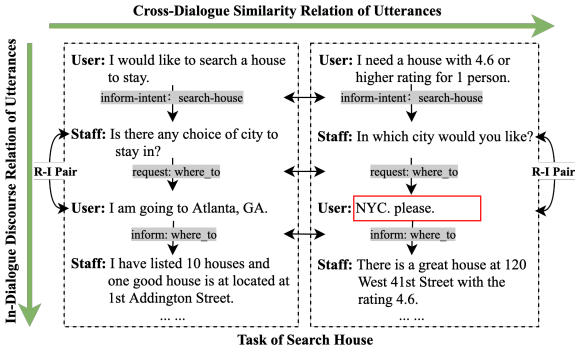

Figure 1: From top to down, the figure shows an example of in-dialogue discourse relation of utterances. From left to right, the figure shows an example of cross-dialogue similarity relation of utterances, the implicit task-related concepts information like "inform-intent:search-house" can be concluded from different dialogues by grouping the utterances with similar semantic.

To address above problems, we proposed a Dialogue Task Clustering Network (DTCN) for TODC. The key points of DTCN are two folds. First, we construct in-dialogue utterance adjacency graph for each dialogue, and encode the graph with graph

[1]https://github.com/Ryan-Lv/DTCN

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4338–4347
November 7–11, 2021.

attention networks (GAT) to build context-aware representations. Second, we cluster all utterances to induce implicit task-related concepts, and then learn utterance cluster representations to utilize this information. Further integrating both kinds of representations into dialogue representations. Finally, training the model with two stages training strategy, which includes pre-training and joint-training stages, the former pretrains a Transformer-based auto-encoder with the proposed Gate-based Transformer decoder for initial clustering assignments, the latter trains jointly the whole model with a self-training strategy for optimizing dialogue representations and dialogue cluster assignments iteratively.

Experimental results on three constructed public dialogue datasets (SGD-S, SGD-M and Multiwoz-T) show that our model significantly outperforms the existing strong text clustering algorithms in all metrics on TODC. Especially, we achieve 19.76% improvement of accuracy on SGD-S dataset compared with the best baseline, which indicates that the proposed dialogue representation method can capture more task-related information.

In summary, the contributions of our paper are as follows:

- We propose an unsupervised Dialogue Task Clustering Network (DTCN). As far as we know, this is the first work on task-oriented clustering for dialogues. Our model learns dialogue representations and clusters dialogues simultaneously by fusing both representations of utterances and utterance clusters.
- We propose a context-aware utterance representation learning model, which uses Graph Attention Network to efficiently capture in-dialogue structural information between utterances and learns the representations with the proposed Gate-based Transformer decoder.
- Experiments on three public datasets show that the proposed model significantly outperforms the existing strong baselines in all metrics on TODC.

2 Related work

Clustering Data representation and clustering algorithm are the two keys to address clustering problems. Previous works (Hartigan, 1979; McLachlan and Basford, 1988; Blei et al., 2001) mainly focused on feature transformation or clustering independently. Data are usually mapped into a feature space and then directly fed into a clustering algorithm to cluster. In the recent years, owing to the development of deep learning, more and more deep clustering methods (Caron et al., 2018; Xie et al., 2016; Guo et al., 2017; Yang et al., 2017; Jiang et al., 2017) were proposed, which can obtain feature representations and cluster assignments simultaneously.

Graph Neural Network Recently, there has been a surge of interest in Graph Neural Networks (GNNs) (Wu et al., 2020b) approaches for graph representation learning. Some GNN variants (Velickovic et al., 2018; Kipf and Welling, 2017) are proposed and also applied in dialogue related tasks. Chen et al. (2020) proposed Graph Attention Matching Network and Recurrent Graph Attention Network based on Graph Attention Network to encode utterances, schema graphs and previous dialogue states. Ghosal et al. (2019) proposed Dialogue Graph Convolutional Network based on Graph Convolutional Network (Kipf and Welling, 2017) to model inter and self-party dependency to improve context understanding.

3 Task formulation

Given an unlabeled dialogue dataset $D = \{d_j\}_{j=1}^{N_{dia}}$, where N_{dia} denotes the total number of dialogues in dataset and $d_j = \{u_i\}_{i=1}^{I}$ denotes one dialogue with I utterances. Task-Oriented Dialogue Clustering (TODC) aims to group D into K_{dia} clusters according to the underlying tasks.

4 The Proposed Model

The proposed Dialogue Task Clustering Network (DTCN) is composed of five modules as shown in Fig.2, and trained with two stages training strategy. In the first stage, we used an autoencoder to learn context-aware utterance representations for initial clustering assignments, in which the Utterance Encoder (UE) and the Structural Context Encoder (SCE) are used as encoder, the Utterance Decoder (UD) module is used as decoder. In the second stage, introducing two new modules based on the pretrained autoencoder, including the Utterance Cluster Representation Learning (UCRL) module for learning utterance cluster representations and the Dialogue Representation Learning (DRL) module for learning dialogue representations, and adopting an iterative training strategy for optimizing jointly dialogue clustering assignments and dialogue representations.

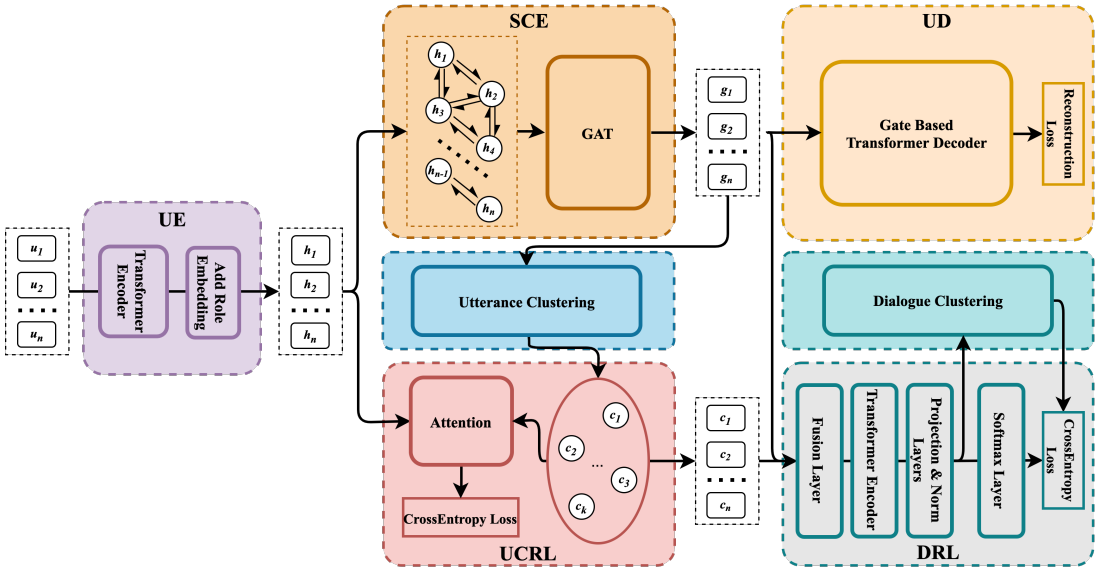

Figure 2: The Frame of Dialogue Task Clustering Network (DTCN).

4.1 Utterance Encoder

UE module aims to encode each utterance to an embedding initially. Specifically, for the i-th utterance $u_i = \{w_t\}_{t=1}^{m_i}$, calculating the word encoding $\epsilon_t \in \mathbb{R}^{d_{mod}}$ for each word w_t firstly as shown in Eq.1,

$$\epsilon_t = emb_t + pos_t \tag{1}$$

where d_{mod} is the embedding size, emb_t is the word embedding and pos_t is the position encoding calculated by the sinusoidal encoding method(Vaswani et al., 2017).

Then, feeding $\{\epsilon_t\}_{t=1}^{m_i}$ into Transformer encoder and adding role embedding $r_i \in \mathbb{R}^{d_{mod}}$ to obtain the initial representation $h_i \in \mathbb{R}^{d_{mod}}$ of the utterance u_i as shown in Eq.2,

$$h_i = Transformer(\epsilon_1, \ldots, \epsilon_{m_i}) + r_i \tag{2}$$

where the mean values of all words of each utterance are used as the outputs of Transformer encoder.

4.2 Structural Context Encoder

SCE module aims to learn context-aware utterance representation, including utterance adjacency graph construction and graph encoding.

Specifically, an adjacency graph $G = (V, E)$ is built for each dialogue $d_j = \{u_i\}_{i=1}^{n_j}$ firstly. $V = \{v_i\}_{i=1}^{n_j}$ is the node set, in which v_i is corresponding to the utterance u_i and its initial representation is h_i. The edges set E between the nodes is defined by N-adjacency relationship as shown in Eq.3,

$$e_{ij} = \begin{cases} 1, & |j-i| \leq N \\ 0, & others \end{cases} \tag{3}$$

where N represents the window size, we suppose that the utterances in the window have discourse relation.

Then, feeding the graph G into Graph Attention Network to obtain the context-aware utterance representation with structural context information as shown in Eq.4,

$$g = GAT(G, h_1, \ldots, h_{n_j}) \tag{4}$$

where $g = \{g_i\}_{i=1}^{n_j}$, $g_i \in \mathbb{R}^{d_{mod}}$ is the improved utterance representation of u_i.

4.3 Utterance Decoder

To better learn context-aware utterance representations, the Gate-based Transformer decoder is proposed as shown in Fig.3. Compared with the standard Transformer decoder, the Gate-based Transformer decoder has an additional Gate-based Extractor sublayer which captures more related information for decoding different words. Specifically, to decode word w_{t+1} in u_i, the Gate-based Extractor sublayer extracts the hidden state g_i^t from g_i through the gate mechanism (Gers, 2001) as shown in Eq.5,

$$g_i^t = g_i \odot sigmoid([a_i^t \| g_i] W^T) \tag{5}$$

where $\cdot\|\cdot$ refers to concatenation, $W^T \in \mathbb{R}^{2d_{mod} \times d_{mod}}$ is a trainable weight matrix, a_i^t is the

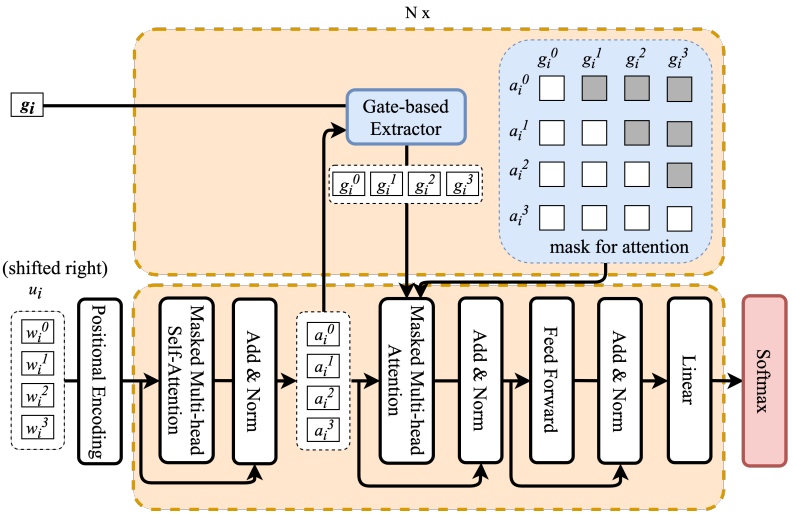

Figure 3: The Gate-based Transformer Decoder.

hidden state corresponding to word w_t through the Masked Multi-head Self-Attention sublayer.

Similar to the Transformer decoder, we use the linear projection and softmax function to convert the outputs to the probability distribution of next word $p_i^{t+1} \in \mathbb{R}^{N_{voc}}$, where N_{voc} is the vocabulary size. The cross-entropy loss $\mathcal{L}_{ud}$ between p_i^{t+1} and the ground-truth word id y_i^{t+1} is employed as shown in Eq.6,

$$\mathcal{L}_{ud} = -\sum_{i=1}^{n_j}\sum_{t=1}^{m_i-1} y_i^{t+1} ln(p_i^{t+1}) \quad (6)$$

where m_i is the number of words in u_i.

4.4 Utterance Cluster Representation Learning

UCRL module first induces implicit task-related concepts by clustering utterances, and then represent this information by leaning representations for each utterance cluster.

Specifically, first grouping all utterances into K_{utt} clusters with the context-aware representations by Gaussian Mixture Model (GMM) (McLachlan and Basford, 1988).

Then, an utterance cluster representation learning method based on the transfer relationship is proposed. It makes use of historical utterance cluster representations and initial representation of the current utterance to update the utterance cluster representation corresponding to the current utterance. Specifically, given a dialogue history $\{u_1, ..., u_{i-1}\}$, let the corresponding utterance cluster representations be $\widetilde{C^{i-1}} = \{c_1, ..., c_{i-1}\}$ and the current utterance representation from UE module be h_i, the utterance cluster representation $c_i^{'}$ corresponding to current utterance u_i is calculated as shown in Eq.7,

$$c_i^{'} = softmax(\frac{h_i \widetilde{C^{i-1}}^T}{\sqrt{d_{mod}}})\widetilde{C^{i-1}} \quad (7)$$

Furthermore, a loss function for cluster representation learning is adopted as shown in Eq.8,

$$\mathcal{L}_{ucrl} = -\sum_{i=1}^{n_j} y_{u_i} ln(softmax(c_i^{'} C^T)) \quad (8)$$

where $C \in \mathbb{R}^{K_{utt} \times d_{mod}}$ is the representation matrix of all utterance clusters, softmax function is used to get the cluster probability distribution, finally cross-entropy between the distribution and the utterance cluster label y_{u_i} corresponding to u_i is calculated.

4.5 Dialogue Representation Learning

DRL module aims to learn dialogue representations by fusing the context-aware utterance representations and the corresponding utterance cluster representations, the former contains the in-dialogue discourse relation information, the latter contains the cross-dialogue task-related concepts information.

Specifically, given a dialogue $d_j = \{u_i\}_{i=1}^{n_j}$, let the corresponding class label be y_{d_j}, the context-aware representation of utterances in the dialogue be $g = \{g_i\}_{i=1}^{n_j}$, the utterance cluster representations be $\widetilde{C} = \{c_i\}_{i=1}^{n_j}$, and the utterance position embeddings be $pos_u = \{pos_{u_i}\}_{i=1}^{n_j}$. The first step

is fusing the g, $\widetilde{C}$, and pos_u into an embedding as shown in Eq.9,

$$\zeta_i = [g_i \| c_i] W^T + pos_{u_i} \tag{9}$$

where $W^T \in \mathbb{R}^{2d_{mod} \times d_{mod}}$, $pos_{u_i} \in \mathbb{R}^{d_{mod}}$ are all trainable.

Then, the Transformer encoder is leveraged to encode $\{\zeta_i\}_{i=1}^{n_j}$, the output o_j is the [CLS] position embedding as shown in Eq.10,

$$o_j = TransformerEncoder(\zeta_1, ..., \zeta_{n_j}) \tag{10}$$

Finally, a Linear layer and a LayerNorm layer project the output o_j into the dialogue representation $z_j \in \mathbb{R}^{K_{dia}}$ as shown in Eq.11,

$$z_j = layernorm(o_j \cdot W^T) \tag{11}$$

A cross-entropy loss as shown in Eq.12, is used to supervise the learning of dialogue representation by maximizing the distance among the dialogue representations in the different classes, which is helpful to cluster dialogues.

$$\mathcal{L}_{cls} = -\sum_{j=1}^{N_{dia}} y_{d_j} ln(softmax(z_j)) \tag{12}$$

4.6 Dialogue Clustering

After obtaining the dialogue representation, we group them into K_{dia} clusters with Gaussian Mixture Model (GMM). And assigning a label for each dialogue, which will be used as pseudo label for training DRL module. Further, a trained DRL module will generates better dialogue representation, and then better dialogue representations help to obtain better clustering assignments. It should be noted that dialogue representations used for the initial clustering is the mean value of utterance representations g from the pretrained autoencoder, and for the subsequent clustering is the learned dialogue representation z.

Due to the instability of the GMM, the initial clustering assignment is obtained by voting after clustering for continuous N_{clu} times as shown in Algo.1.

4.7 Model Training Process

A two-stage training strategy including pre-training and joint-training is employed for model training. In the pre-training stage, learning the context-aware utterance representation g_i for each utterance with the autoencoder based on encoder-decoder architecture for initial clustering assignments, where UE, SCE modules as the encoder and UD as the decoder. The pre-training loss is defined by Eq.13,

$$\mathcal{L}_{Pre} = \mathcal{L}_{ud} \tag{13}$$

In the joint-training stage, an iterative training strategy is adopted. In each iteration, the label reassignment strategy is employed to improve the confidence of clustering assignments. Specifically, clustering all utterances and dialogues after each training epoch, and updating the old clustering assignment by best mapping the clusters between new and old clustering assignment using the Hungarian algorithm (Kuhn, 1955). However, the pseudo labels used by UCRL and DRL modules are not assigned immediately, but reassigning every $interval$ epochs. Finally, stopping training when the change between two consecutive dialogue clustering assignment is less than $tol\%$ or reaching the maximum training epochs max_{ep}. The last dialogue clustering assignment is used as the final clustering results. The loss is defined by Eq.14,

$$\mathcal{L}_{Joint} = \mathcal{L}_{ud} + \beta \cdot \mathcal{L}_{ucrl} + \mathcal{L}_{cls} \tag{14}$$

where β is loss coefficient.

Algorithm 1: Initial labels assignment algorithm

Input : Initial count matrix $\theta = \mathbf{0}$; Initial labels assignment vector $A = \mathbf{0}$; Clustering assignments sequence $\{A_n\}_{n=1}^{N_{clu}}$.

```
for i ∈ {1, 2, 3, ..., N_clu} do
    if i == 1 then
        A = A_1
    end
    Best map: mapping assignment A_i to
              assignment A using Hungarian
              algorithm.
    Update θ: θ_ij = θ_ij + 1 if s_i is assigned to
              cluster j in mapped assignment A_i.
    Update A: A_i = argmax(θ_i) where θ_i is
              i-th row of θ.
end
Return final labels assignment vector A;
```

5 Experiments

5.1 Datasets

In order to better evaluate the performance of different algorithms on TODC, we constructed three public task-divided dialogue datasets based on Schema-Guided Dialogue (SGD) (Rastogi et al., 2020) and Multiwoz dataset (Zang et al., 2020). In both SGD and Multiwoz datasets, we determine whether two

Datasets	SGD-S	SGD-M	Multiwoz-T
Tasks Number	29	59	35
Dialogues Number	3925	4722	9695
Dialogues Length(avg)	15.57	21.68	13.94

Table 1: The statistics of the datasets.

dialogues belong to the same dialogue task by judging whether the two dialogues contain the same set of active-intents. Finally, three datasets labeled by dialogue task are constructed: SGD-S includes-single domain dialogues of SGD dataset, SGD-M includes multiple-domains dialogues of SGD dataset, Multiwoz-T includes all dialogues of Multiwoz dataset. Detailed division instructions and datasets will be released. Tab.1 shows the statistics of them.

5.2 Implement Details

In our experiments, the hidden size is set to 256. Using 3-layers Transformer encoder for both UE and DRL module, 2-layers GAT for SCE module, and 3-layers Gated-based Transformer decoder for UD module. The window size is set to 2, 2, 1 for SGD-S, SGD-M and Multiwoz-T respectively.

In the pre-training stage, we train 100 epochs with batch size 16 on each dataset with the same optimizer settings as the Transformer (Vaswani et al., 2017). In the joint-training stage, estimating K_{utt} by BIC score (Schwarz et al., 1978) to 50, 50, 60, and setting $interval$ to 2, 2, 1 for SGD-S, SGD-M and Multiwoz-T datasets respectively. Besides, setting the coefficient of $\mathcal{L}_{ucrl}$ to 10 for stabilizing the utterance cluster representations quickly. And the learning rate at each step is calculated as shown in Eq.15,

$$lr = lr_{max} \cdot wm_{stp}^{0.5} \cdot min(N_{stp}^{-0.5}, N_{stp} \cdot wm_{stp}^{-1.5}) \quad (15)$$

where the maximum learning rate lr_{max} is set to 1e-3, 1e-3 and 5e-4 for SGD-S, SGD-M and Multiwoz-T respectively, the warmup steps $wm_{stp} = interval \cdot N_{dia\_bth}$, N_{dia_bth} is the batch number of the corresponding dataset. Such warmup strategy increases learning rate linearly between the first and second dialogue labels reassignment until lr_{max}, then decreases proportionally.

5.3 Baselines and Metrics

Three types of baselines are adopted to be compared with the proposed model on TODC performance.

Raw feature based models. Using bag of words model (BOW) and TF-IDF feature to represent dialogues, and clustering with *LDA* (Blei et al., 2001), *K-means* and *GMM* algorithms respectively.

Pretrained feature based models. Representing dialogues with the mean values of all utterance representations extracted from official pretrained *SkipThought* (Kiros et al., 2015), *TODBert* (Wu et al., 2020a) and *SentenceBert* (Reimers and Gurevych, 2019) models, then cluster with *GMM*.

Deep clustering models. Four popular deep clustering models are adopted as strong baselines. *DCN* (Yang et al., 2017) used k-means clustering loss to learn clustering friendly representations. *VaDE* (Jiang et al., 2017) is a generative deep clustering model based on variational autoencoder. *DEC* (Xie et al., 2016) designed a clustering objective to guide the learning of the data representations. *IDEC* (Guo et al., 2017) is a modified version of *DEC* with a reconstruction loss to preserve local structure.

We run 5 times continuously for all baselines, and then report the mean value and standard deviation. For *DCN*, *VaDE*, *DEC* and *IDEC*, using the same settings as the previous works, and searching the update interval in {2, 5, 10, 20}. In addition, searching the γ in {1.0, 0.1, 0.01, 0.001} for *IDEC*, and λ in {0.001, 0.005, 0.01, 0.05, 0.1, 0.5} for *DCN*.

Metrics Four popular metrics are adopted to evaluate TODC performance, including Accuracy (ACC), Purity, Normalized Mutual Information (NMI) (Strehl and Ghosh, 2002), and Adjusted Rand Index (ARI) (Hubert and Arabie, 1985). For each metric, a larger value implies a better clustering performance.

5.4 Main Results

Tab.2 shows the clustering performance of different methods. We can see that the proposed model outperforms all three types baselines significantly. Compared with the best baseline, our model improves ACC by 19.76%, 16.67% and 4.87%, Purity by 12.83%, 14.80% and 7.26%, NMI by 9.45%, 8.65% and 3.45%, ARI by 22.59%, 20.49% and 7.10% on SGD-S, SGD-M and Multiwoz-T datasets respectively. The results show that the obtained dialogue representations can capture more task-related information.

Model	SGD-S				SGD-M				Multiwoz-T			
	ACC	Purity	NMI	ARI	ACC	Purity	NMI	ARI	ACC	Purity	NMI	ARI
Raw Feature Models												
K-means	59.04±1.41	70.32±1.50	80.69±0.58	55.60±2.12	50.76±3.52	55.53±3.26	73.25±1.83	45.55±3.29	29.44±1.79	37.38±1.28	45.65±1.05	24.07±1.90
GMM	59.96±3.05	71.18±1.59	81.18±1.09	57.39±3.81	50.06±3.50	55.12±2.85	73.20±1.45	45.82±3.07	34.90±1.88	42.51±1.53	50.19±1.68	28.88±2.61
LDA	55.78±2.64	68.0±1.31	75.13±0.72	50.06±2.08	39.35±2.03	40.34±1.64	67.94±0.98	33.12±1.81	39.58±1.14	49.32±0.67	54.66±0.42	33.36±0.40
PreTrained Feature Models												
SkipThought+GMM	36.16±2.16	44.56±0.94	50.41±2.14	30.43±2.83	15.59±0.95	17.98±0.72	31.37±0.95	7.50±0.55	14.35±0.51	22.99±0.30	23.84±0.37	6.85±0.31
TODBert+GMM	54.93±2.79	71.23±1.44	77.87±1.00	52.32±3.69	53.78±0.79	69.00±1.46	77.11±0.98	53.76±2.24	32.56±1.64	43.50±1.84	53.93±2.91	30.44±1.40
SentenceBert+GMM	57.18±3.64	69.88±1.63	78.62±0.89	55.37±3.72	55.85±1.88	63.18±1.44	76.69±0.98	50.36±2.34	37.97±2.41	49.86±2.27	59.45±2.32	30.62±2.58
Deep Clustering Models												
DCN	55.32±2.73	64.29±1.37	79.09±1.62	53.28±3.09	42.08±2.33	49.99±3.39	73.57±1.39	34.93±2.29	47.08±0.32	60.04±0.31	68.39±0.39	41.71±0.52
VaDE	60.66±1.55	70.97±1.67	79.24±0.67	54.67±1.22	51.55±3.49	54.14±3.65	74.62±0.96	45.71±2.54	50.51±0.14	56.36±0.15	66.55±0.07	44.45±0.19
DEC	64.10±1.53	73.69±0.90	82.95±0.48	59.28±1.27	68.89±2.20	70.44±2.31	87.89±0.99	65.24±2.73	56.16±1.66	65.12±0.85	78.43±0.52	51.58±1.61
IDEC	64.61±1.37	75.15±0.51	82.92±0.22	59.17±1.35	70.32±1.10	77.03±0.74	88.03±0.45	67.77±1.08	58.64±3.19	66.03±1.21	78.39±1.34	54.06±2.35
Our Model												
PreTrained	66.67±3.11	79.02±1.55	85.46±1.13	63.65±2.68	49.84±1.59	53.09±1.94	69.39±2.04	45.33±2.63	45.71±2.16	57.68±1.21	67.27±1.00	39.44±2.12
Full model	**84.37±2.33**	**87.98±2.31**	**92.37±0.93**	**82.76±2.84**	**86.99±2.14**	**91.83±1.19**	**96.68±0.51**	**88.26±1.97**	**63.51±3.03**	**73.29±2.60**	**81.84±1.49**	**61.16±2.41**

Table 2: Comparison of clustering performance on three datasets.

5.5 Ablation Studies

Ablation Studies of Major Modules We conduct ablation studies to evaluate the compact of different components in our model.

Model	SGD-S			
	ACC	Purity	NMI	ARI
Full	**84.37±2.33**	**87.98±2.31**	**92.37±0.93**	**82.76±2.84**
-w/o SCE	64.67±2.84	72.63±3.12	80.41±30.3	62.76±2.64
-w/o UCRL	75.18±1.01	84.34±0.17	89.64±0.36	74.39±1.07

Table 3: Ablation studies for major modules.

As shown in Tab.3, both SCE and UCRL contribute to the proposed model, and compared to UCRL module, the SCE module has a greater impact on performance. On the one hand, it shows that the integration of the structural context of utterance can improve the quality of utterance representation and further affect the dialogue representation. On the other hand, the utterance clustering assignment based on the utterance representation from SCE module has a direct impact on UCRL module, and the utterance cluster representations from UCRL module will directly affect dialogue representation.

Ablation Studies of Losses We also conduct ablation studies to evaluate the compact of different losses in our model.

Model	SGD-S			
	ACC	Purity	NMI	ARI
Full	**84.37±2.33**	**87.98±2.31**	**92.37±0.93**	**82.76±2.84**
PreTrained	66.67±3.11	79.02±1.55	85.46±1.13	63.65±2.68
-w/o $\mathcal{L}_{ud}$	79.86±3.21	86.02±1.60	89.64±1.14	77.03±3.00
-w/o $\mathcal{L}_{ucrl}$	75.16±2.12	85.03±0.50	89.70±0.34	75.38±3.52

Table 4: Ablation studies for losses. "-w/o $\mathcal{L}_{ucrl}$" refers to the performance using the utterance cluster embedding simply without learning by the $\mathcal{L}_{ucrl}$.

As shown in Tab.4, the ACC is reduces by 9.21% after removing $\mathcal{L}_{ucrl}$ loss, which indicates that the structure information learned through the logical transfer relationship between utterance clusters is helpful to distinguish different tasks. And the ACC is reduced by 4.51% after removing $\mathcal{L}_{ud}$ loss, which indicates that stabilizing the utterance representations helps stabilize model training. Meanwhile, we can see that all performance have been significantly improved after joint training, which indicates that introducing the induced implicit concepts information by clustering utterances and adopting an iterative training strategy are beneficial for TODC.

5.6 Window Size N Analysis

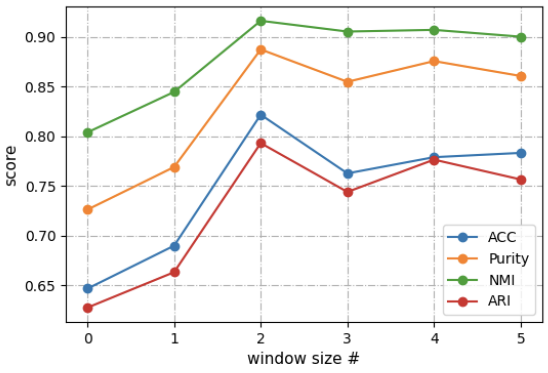

Figure 4: The performance with different window size.

We analyze the impact of the window size N on all performance on SGD-S dataset. As shown in Fig.4, as the window size increases, the performance is significantly improved. When the window size is 2, all performance reaches the maximum, then decreases slightly and stabilizes. This indicates that the optimal window size is 2. If the size of window is too small, the context information in-

troduced is insufficient, and if it is too large, it will introduce too much noise and affect performance.

5.7 Clustering Number K_{dia} Analysis

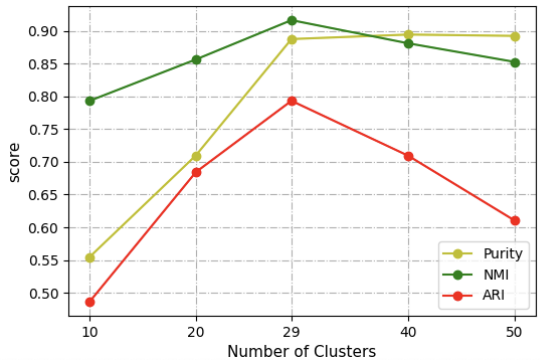

Figure 5: The performance of Purity, NMI and ARI with different dialogue clustering number.

We also analyze the impact of the clustering number on performance of NMI, ARI and Purity on SGD-S dataset. As shown in Fig.5, as the clustering number increases, all performance is significantly improved. When the clustering number is 29, which is the ground-truth number of tasks in SGD-S dataset, all performance reaches the maximum. Then the performance of Purity stabilizes, while NMI and ARI decrease.

The Purity measures the degree to which the samples in the cluster belong to the same true category. As the number of clusters increases, the purity will gradually increase as shown in the Fig.5, and then stabilize. NMI and ARI measure the degree of overlap between clustering and true category distributions. When the clusters number differs greatly from the true categories number, the performance will significantly decrease as shown in the Fig.5.

6 Case Studies

We selected some typical utterance clusters to evaluate the quality of learned context-aware utterance representation, and analysis whether some interpretable task-related concepts can be induced by obtained utterance clusters.

One case is shown in the Fig.6, obviously, these utterances are all about the concept of "user wants to search one-way flight". Besides, we also found some special utterance clusters. Another case is shown in Fig.7, there are two segments from two dialogues, to our surprise, all of these utterances are grouped into one cluster. This phenomenon can be explained from two aspects. First, from

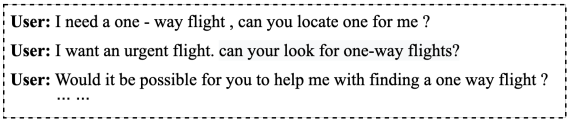

Figure 6: An example of utterance cluster, these utterances are about the concept of "SearchOneWayFilght"

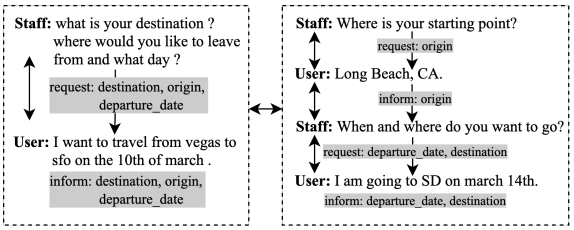

Figure 7: An example of special utterance cluster, these utterances are all related to slots of "destination, origin and departure_date".

top to down, discourse relation information and the involved slots information make the representations of related utterances similar. For example, the both utterances of left segment are involved the same three slots, and associated by request-inform pair relationship, the both representations tend to be similar after incorporating this information. Further, from left to right, the cross-dialogue utterances with the similar context information are grouped automatically.

These indicate that our model can fully integrate contextual information by constructing adjacency graphs, and can also induce interpretable concepts through clustering utterances.

7 Conclusion and Future Work

This paper proposes a Dialogue Task Clustering Network for dialogue task clustering. The model makes use of both in-dialogue discourse relation information and cross-dialogue utterance similarity relation information to build dialogue representations. And an end-to-end iterative strategy of jointly dialogue representation learning and clustering is used to train the model. Experiments on three public datasets show that the proposed model significantly outperforms the existing strong clustering algorithms on dialogue task clustering. In the future, we will further induce more detailed task-related concepts information and explore the inner structure in each dialogue cluster for task induction and definition.

Acknowledgments

We would like to thank anonymous reviewers for their suggestions and comments. The work was supported by the National Natural Science Foundation of China (NSFC62076032)

References

Prafulla Bafna, Dhanya Pramod, and Anagha Vaidya. 2016. Document clustering: Tf-idf approach. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 61–66. IEEE.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2001. Latent dirichlet allocation. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 601–608. MIT Press.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7521–7528.

Felix Gers. 2001. *Long short-term memory in recurrent neural networks*. Ph.D. thesis, Verlag nicht ermittelbar.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. DialogueGCN: A graph convolutional neural network for emotion recognition in 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 154–164, Hong Kong, China. Association for Computational Linguistics.

Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. 2017. Improved deep embedded clustering with local structure preservation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1753–1759. ijcai.org.

John A Hartigan. 1979. A k-means clustering algorithm: Algorithm as 136. *Appl. Stat.*, 28:126–130.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. 2017. Variational deep embedding: An unsupervised and generative approach to clustering. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1965–1972. ijcai.org.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3294–3302.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

GJ McLachlan and KE Basford. 1988. Mixture models marcel dekker. *New York*.

Joel Larocca Neto, Alexandre D Santos, Celso AA Kaestner, Neto Alexandre, D Santos, et al. 2000. Document clustering and text summarization.

Aytug Onan, Hasan Bulut, and Serdar Korukoglu. 2017. An improved ant algorithm with lda-based representation for text document clustering. *Journal of Information Science*, 43(2):275–292.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Gideon Schwarz et al. 1978. Estimating the dimension of a model. *Annals of statistics*, 6(2):461–464.

Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance rewriter. *arXiv preprint arXiv:1906.07004*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Chien-Sheng Wu, Steven CH Hoi, Richard Socher, and Caiming Xiong. 2020a. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020b. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*.

Junyuan Xie, Ross B. Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 478–487. JMLR.org.

Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3861–3870. PMLR.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. *arXiv preprint arXiv:2007.12720*.