Task-Oriented Clustering for Dialogues

Chenxu Lv, Hengtong Lu, Shuyu Lei, Huixing Jiang, Wei Wu, Caixia Yuan, Xiaojie Wang
Beijing University of Posts and Telecommunications, Beijing, China
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 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.

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
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.

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.
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,

$$\epsilon_t = emb_t + pos_t$$  

(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.

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

(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,

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

(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,

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

(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 $g'_i$ from $g_i$ through the gate mechanism (Gers, 2001) as shown in Eq.5,

$$g'_i = g_i \odot \text{sigmoid}(|a'_i||g_i|W^T)$$  

(5)

where $\cdot||\cdot$ refers to concatenation, $W^T \in \mathbb{R}^{2d_{mod} \times d_{mod}}$ is a trainable weight matrix, $a'_i$ is the
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 \( p_{t+1}^i \in \mathbb{R}^{N_{voc}} \), where \( N_{voc} \) is the vocabulary size. The cross-entropy loss \( L_{ud} \) between \( p_{t+1}^i \) and the ground-truth word id \( y_{t+1}^i \) is employed as shown in Eq.6,

\[
L_{ud} = - \sum_{i=1}^{m_i} \sum_{t=1}^{y_{t+1}^i} y_{t+1}^i \ln(p_{t+1}^i) \tag{6}
\]

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 \( C_{i-1} = \{c_1, ..., c_{i-1}\} \) 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,

\[
c_i = \text{softmax}(h_i C_{i-1}^T \sqrt{d_{mod}}) C_{i-1} \tag{7}
\]

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

\[
L_{ucrl} = - \sum_{i=1}^{n_j} y_{u_i} \ln(\text{softmax}(c_i^T C)) \tag{8}
\]

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 \( g_j = \{g_i\}_{i=1}^{n_j} \), the utterance cluster representations be \( C = \{c_i\}_{i=1}^{n_j} \), and the utterance position embeddings be \( pos_u = \{pos_u_i\}_{i=1}^{n_j} \). The first step
is fusing the $g$, $\tilde{C}$, and $\text{pos}_u$ into an embedding as shown in Eq.9,

$$
\zeta_i = [g_i]\cdot c_i + \text{pos}_u \tag{9}
$$

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

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

$$
o_j = \text{TransformerEncoder}(\zeta_1, ..., \zeta_n) \tag{10}
$$

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,

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

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.

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

### 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 $N_{clus}$ times as shown in Alg.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, SGE 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 re-assignment 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 $\beta$ is loss coefficient.

### 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
Table 1: The statistics of the datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>SGD-S</th>
<th>SGD-M</th>
<th>Multiwoz-T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks Number</td>
<td>29</td>
<td>59</td>
<td>35</td>
</tr>
<tr>
<td>Dialogues Number</td>
<td>3925</td>
<td>4722</td>
<td>9695</td>
</tr>
<tr>
<td>Dialogues Length(avg)</td>
<td>15.57</td>
<td>21.68</td>
<td>13.94</td>
</tr>
</tbody>
</table>

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 $L_{uret}$ to 10 for stabilizing the utterance cluster representations quickly. And the learning rate at each step is calculated as shown in Eq.15,

$$lr = lr_{max} \cdot w_{stp}^{0.5} \cdot \text{min}(N_{stp}^{-0.5}, N_{stp} \cdot w_{stp}^{-1.5})$$

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 $w_{stp} = \text{interval} \cdot N_{dia_{lth}}$, $N_{dia_{lth}}$ 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 $\gamma$ in {1.0, 0.1, 0.01, 0.001} for IDEC, and $\lambda$ 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.
We conduct Ablation Studies of Major Modules in our model. Table 4: Ablation studies for losses. **w/o** SCE means 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 $L_{ud}$ loss, which indicates that the utterance representations helps stabilize model training.

### 5.5 Ablation Studies

#### Ablation Studies of Major Modules

We conduct ablation studies to evaluate the compact of different components in our model.

<table>
<thead>
<tr>
<th>Model</th>
<th>ACC</th>
<th>Purity</th>
<th>NMI</th>
<th>ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGD-S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-means</td>
<td>70.31±1.50</td>
<td>80.61±0.58</td>
<td>55.60±2.12</td>
<td>37.32±1.63</td>
</tr>
<tr>
<td>GMM</td>
<td>78.33±1.89</td>
<td>73.20±1.45</td>
<td>45.82±3.07</td>
<td>34.90±1.88</td>
</tr>
<tr>
<td>LDC</td>
<td>78.33±1.89</td>
<td>73.20±1.45</td>
<td>45.82±3.07</td>
<td>34.90±1.88</td>
</tr>
<tr>
<td>PreTrained Feature Models</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SkipThought+GMM</td>
<td>44.86±1.40</td>
<td>50.41±2.14</td>
<td>30.43±2.87</td>
<td>17.98±0.72</td>
</tr>
<tr>
<td>OD+GMM</td>
<td>52.32±3.00</td>
<td>59.20±2.60</td>
<td>29.57±3.17</td>
<td>15.59±0.95</td>
</tr>
<tr>
<td>Sentence+GMM</td>
<td>57.18±3.64</td>
<td>69.88±1.44</td>
<td>78.12±0.98</td>
<td>39.37±2.32</td>
</tr>
<tr>
<td>Deep Clustering Models</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCN</td>
<td>64.32±2.73</td>
<td>70.94±1.62</td>
<td>53.28±3.09</td>
<td>42.08±2.33</td>
</tr>
<tr>
<td>VaDE</td>
<td>60.46±1.55</td>
<td>70.97±1.67</td>
<td>54.67±2.12</td>
<td>51.55±3.49</td>
</tr>
<tr>
<td>DEC</td>
<td>64.10±3.53</td>
<td>73.69±1.90</td>
<td>59.28±1.27</td>
<td>48.89±2.20</td>
</tr>
<tr>
<td>DEC</td>
<td>64.10±3.53</td>
<td>73.69±1.90</td>
<td>59.28±1.27</td>
<td>48.89±2.20</td>
</tr>
<tr>
<td>Our Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PreTrained</td>
<td>66.87±1.11</td>
<td>70.92±1.55</td>
<td>65.46±1.13</td>
<td>63.65±2.00</td>
</tr>
<tr>
<td>Full model</td>
<td>84.37±2.33</td>
<td>87.98±2.31</td>
<td>92.37±0.93</td>
<td>82.76±2.84</td>
</tr>
</tbody>
</table>

#### Ablation Studies of Losses

As shown in Tab.4, the ACC is reduces by 9.21% after removing $L_{ucrl}$ 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 $L_{ud}$ loss, which indicates that the utterance representations helps stabilize model training.

### 5.6 Window Size $N$ Analysis

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-
introduced 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.](image)

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 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


