Dialogue State Distillation Network with Inter-slot Contrastive Learning for Dialogue State Tracking

Jing Xu†, Dandan Song††, Chong Liu‡, Siu Cheung Hui§, Fei Li¶, Qiang Ju‡, Xiaoran He‡, Jian Xie††

1School of Computer Science & Technology, Beijing Institute of Technology, Beijing, China
2Baidu Inc., Beijing, China
3Nanyang Technological University, Singapore

{xujing, sdd}@bit.edu.cn, {liuchong06, lifei21, juqiang, hexiaonan, xiejian01}@baidu.com, asschui@ntu.edu.sg

Abstract

In task-oriented dialogue systems, Dialogue State Tracking (DST) aims to extract users’ intentions from the dialogue history. Currently, most existing approaches suffer from error propagation and are unable to dynamically select relevant information when utilizing previous dialogue states. Moreover, the relations between the updates of different slots provide vital clues for DST. However, the existing approaches rely only on predefined graphs to indirectly capture the relations. In this paper, we propose a Dialogue State Distillation Network (DSDN) to utilize relevant information of previous dialogue states and migrate the gap of utilization between training and testing. Thus, it can dynamically exploit previous dialogue states and avoid introducing error propagation simultaneously. Further, we propose an inter-slot contrastive learning loss to effectively capture the slot co-update relations from dialogue context. Experiments are conducted on the widely used MultiWOZ 2.0 and MultiWOZ 2.1 datasets. The experimental results show that our proposed model achieves the state-of-the-art performance for DST.

Introduction

As a key component of task-oriented dialogue systems, Dialogue State Tracking (DST) aims to extract users’ goals or intents over the continuation of multiple turns of a dialogue, which are described as a compact dialogue state. Specifically, a DST model aims to extract the corresponding values of pre-defined slots given a dialogue history. Thus, the extracted dialogue state can be represented as a set of (slot, value) pairs. A DST example is shown in Figure 1.

As previous dialogue states can provide effective and explicit information of a dialogue history (Kim et al. 2020), recent approaches attempt to utilize the previous dialogue state information for DST. Most of these approaches (Zhu et al. 2020; Zeng and Nie 2020a,b) concatenate the dialogue context and the corresponding previous dialogue state as input. However, the golden dialogue states can only be obtained during training. Therefore, utilizing the predicted dialogue states may lead to error propagation during the testing process. To alleviate this problem, gated recurrent units are used in a graph attention network to learn to decide which dialogue states are kept (Chen et al. 2020a). In addition, a uniform scheduled sampling strategy is also used to make the DST model adapt to the data noise of dialogue states, thus improving the robustness of the model (Zhou et al. 2021). Unfortunately, these attempts are still unable to tackle the problems of the unavailable golden dialogue states and the potential error of predicted dialogue states, though they strive to reduce the negative impact of wrongly predicted dialogue states during the testing process. Moreover, for different current dialogue contexts, the relevant information of previous dialogue states should be dynamically obtained in contrast to missing in the current approaches. Therefore, it is a challenging problem on how to dynamically incorporate previous dialogue states without error propagation for DST.

Figure 1: An example of DST. Utterances on the left side are from the user and system over the continuation of the multiple turns of a dialogue. The corresponding dialogue state for each turn is on the right side. Specifically, “train-day: Friday” represents that the value of the slot “day” in the domain “train” is “Friday”.

*Work done during internship at Baidu Inc.
††Corresponding authors.
© 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
Thus, the values of the slots “book day” and “book stay” are usually updated at the same time in the domain “hotel”. Although several recent approaches incorporate the slot relations across domains to indirectly capture the slot co-update relations for DST (Chen et al. 2020a; Zeng and Nie 2020b; Zhu et al. 2020; Feng et al. 2022), they rely on predefined schema graphs or designed graphs and fail to capture the relations from dialogue context automatically. Recent works (Ye et al. 2021; Wang et al. 2022b) also attempt to use self-attention to capture slot dependency. However, they mainly provide mutual guidance among slots in an implicit manner. Therefore, how to explicitly capture the slot co-update relations without predefined graphs is another important issue for DST.

In this paper, we propose an effective model equipped with a Dialogue State Distillation Network (DSDN) to dynamically exploit previous dialogue states without introducing error propagation. In DSDN, a combination of turn-level and dialogue-level slot relevant information is first encoded by an encoder. Then, a teacher network with golden previous dialogue states tackles SOP while a student network with slot only concatenation learns to imitate the behavior of the teacher network. As a result, the student network can utilize the previous dialogue states without extra input. Moreover, the rich interactions between the encoder and the teacher/student network are conducted to push the teacher/student network to dynamically decide the utilization of previous dialogue states. Finally, the well-learned student network can help our model tackle DST more effectively.

Additionally, an inter-slot contrastive learning loss is also proposed to explicitly capture the slot co-update relations. In details, we construct the contrastive learning loss between slot representations from both turn-level and dialogue-level according to the labels of SOP. In this way, the model can directly capture the label correlations of SOP between slots automatically and then help improve DST performance. In summary, the main contributions of this paper are as follows:

- We propose an effective Dialogue State Distillation Network to dynamically exploit previous dialogue states without introducing error propagation. To the best of our knowledge, this study is the first to exploit knowledge distillation for DST.
- We propose an inter-slot contrastive learning loss to capture the slot co-update relations from dialogue context automatically.
- Extensive experiments are conducted on the widely used MultiWOZ 2.0 and MultiWOZ 2.1 datasets. Our proposed model achieves the state-of-the-art performance on MultiWOZ 2.1 and competitive performance on MultiWOZ 2.0.

Related Work

Dialogue State Tracking In task-oriented dialogue systems, as natural language understanding (Wang et al. 2022a; Xing and Tsang 2022) and natural language generation (Mi, Wang, and Li 2022) are standalone and less interactive with each other, recent works focus more on dialogue management such as dialogue state tracking (Zhang et al. 2020c). Existing approaches can be divided into two categories: ontology-based and ontology-free. The ontology-based approaches (Zhong, Xiong, and Socher 2018; Ramadan, Budzianowski, and Gasic 2018; Ye et al. 2021) require an ontology including all possible predefined values for each slot and are simplified into a multi-class classification task. Conversely, the ontology-free approaches (Xu and Hu 2018; Hosseini-Asl et al. 2020; Gao et al. 2019; Feng et al. 2022) generate the values of slots from dialogue context or vocabularies. In this paper, our work is related to the ontology-based approaches.

Recently, the utilization of previous dialogue states has attracted much attention for DST. Several works (Kim et al. 2020; Zhu et al. 2020; Zeng and Nie 2020a,b) use the previous dialogue states as part of the input. Once a slot in the dialogue states is wrongly predicted, the error information will continue propagating until the end of the dialogue. To improve the utilization effectiveness, the previous dialogue state information is controlled by a recurrent graph attention network (Chen et al. 2020a). A multi-level fusion gate and a uniform scheduled sampling strategy are also used to reduce the negative impact of wrong predictions (Zhou et al. 2021).

As for slot co-update relations, most existing approaches (Zhu et al. 2020; Chen et al. 2020a; Wu et al. 2020; Ouyang et al. 2020; Lin, Tseng, and Byrne 2021; Feng et al. 2022) use graph neural networks based on predefined schema graphs to exploit the prior knowledge of domain or slot dependency, thus capturing them indirectly. Among these models, the co-update relations between slots are only explicitly defined by Feng et al. (2022). They fuse schema graphs and dialogue context to obtain diverse dynamic slot relations which include the slot co-update relations. The obtained slot relations further enhance the final predictions.

Knowledge Distillation Knowledge distillation (Hinton, Vinyals, and Dean 2015) aims to distill knowledge from teacher networks and transfer it to student networks. Traditionally, it pushes the student networks to imitate the feature representations or probability distributions of the teacher networks. Recently, knowledge distillation has been widely used in natural language processing (Sun et al. 2020; Zhang et al. 2020b; Wei et al. 2021; Zhou, Xu, and McAuley 2022). In this paper, we propose a novel knowledge distillation network to migrate the gap of utilizing previous dialogue states between the training and testing processes.

Contrastive Learning Contrastive learning is first used in computer vision with self-supervised learning (Chen et al. 2020b), which increases the similarity between the representations of original data samples and differently augmented data samples. Further, supervised contrastive learning (Khosla et al. 2020) extends it and generates positive samples from not only data augmentation but also training instances with the same label. In natural language processing, contrastive learning has been widely used in diverse fields, such as pre-trained language model and information extraction (Wang et al. 2021a,b; Das et al. 2022; Saha, Yadav, and Bansal 2022). As for DST, we are the first to introduce contrastive learning to exploit the slot co-update relations from the whole dialogue automatically.
Proposed Model

For DST, we use $X =\{(U_1, R_1), \ldots , (U_T , R_T )\}$ to represent a dialogue with $T$ turns and $S = \{s_1, \ldots , s_J\}$ to represent a pre-defined slot set, where $U_t$ is the user utterance at the $t$-th dialogue turn, $R_t$ denotes the corresponding system response and $J$ denotes the total number of pre-defined slots. For each turn $t$, given $X =\{(U_1, R_1), \ldots , (U_t, R_t)\}(1 \leq t \leq T)$, the goal of DST is to extract the dialogue state which can be represented as $B =\{(s_1,v_1), \ldots , (s_J,v_J)\}$, where $v_{t_j}$ denotes the value of $s_j$ for the $t$-th turn. Note that if a slot is not mentioned, its value will be annotated as “none”. As for any slot, we follow (Wang et al. 2022b) to use a concatenation of its domain name and slot name to represent it, thus integrating both domain information and slot information.

In this section, we propose a novel model to tackle DST. Figure 2 shows the proposed model which consists of four modules: Encoder, Dialogue State Distillation, Inter-slot Contrastive Learning and Decoder. Except the Inter-slot Contrastive Learning module, the other three modules constitute the Dialogue State Distillation Network. Next, we will introduce each module in details.

Encoder

Based on previous works (Lee, Lee, and Kim 2019; Ye et al. 2021; Wang et al. 2022b), we aim to combine turn-level and dialogue-level slot relevant information in this module. First, we use a pre-trained BERT (Kenton and Toutanova 2019) to encode dialogue context, slots and slot values respectively. Then, we retrieve slot relevant context from dialogue at turn-level and dialogue-level. Finally, a max-pooling operation is used to fuse the retrieved information.

Base Encoder

For the dialogue context at the $t$-th turn, we concatenate the user utterance and the system utterance, and encode them into hidden representations:

$$ R_t = \text{BERT}_{\text{finetune}}([\text{CLS}] \oplus U_t \oplus [\text{SEP}] \oplus R_t \oplus [\text{SEP}]) $$

(1)

where $[\text{CLS}]$ and $[\text{SEP}]$ denote special tokens in BERT, and $\text{BERT}_{\text{finetune}}$ denotes that the used BERT will be finetuned during the training process. For any given slot $s_j \in S$ or the corresponding value $v_{t_j}$ at the given $t$-th turn, we use another BERT to encode it, whose parameters are fixed during the training process. Unlike encoding dialogue context, we use the hidden representation of the special token $[\text{CLS}]$ to represent $s_j$ or $v_{t_j}$:

$$ h^{S_j} = \text{BERT}_{\text{fixed}}([\text{CLS}] \oplus s_j \oplus [\text{SEP}]) $$

(2)

$$ h^{V_{t_j}} = \text{BERT}_{\text{fixed}}([\text{CLS}] \oplus v_{t_j} \oplus [\text{SEP}]) $$

(3)

Multi-level Slot Attention

Based on the hidden representations of dialogue context and slots, we further utilize turn-level and dialogue-level slot attention to obtain slot relevant information. A multi-head attention mechanism (Vaswani et al. 2017) is first used for the turn-level slot attention and we use MultiHead($Q, K, V$) to represent it, where $Q$ denotes a query matrix or vector, $K$ and $V$ denote a key matrix and a value matrix respectively. Thus, the turn-level slot attention is denoted as follows:

$$ r^{S_j}_t = \text{MultiHead}(h^{S_j}, R_t, R_t) $$

(4)

As the hidden representations of current dialogue context at the $t$-th turn encoded by Base Encoder lack of previous dialogue context information, we use a transformer encoder (Vaswani et al. 2017) to aggregate contextual information before dialogue-level slot attention. It is composed of $N$ stacked layers and each layer consists of two sub-layers including a multi-head attention mechanism and a position-wise fully connected feed-forward network. We use the output of the last layer as the contextual representations of each slot, which is denoted as follows:

$$ D^{S}_t = \text{TransformerEncoder}(r^{S}_{1:t}, \ldots , r^{S}_{t}) $$

(5)

After that, the dialogue-level slot attention is applied to the contextual representations $D^{S}_t$, thus summarizing the slot relevant information from the up-to-now utterances for the slot $s_j$. Similarly, we use another multi-head attention mechanism as follows:

$$ d^{S_j}_t = \text{MultiHead}(h^{S_j} \cdot D^{S}_t \cdot D^{N}_t) $$

(6)

Fusion

To fuse the turn-level and dialogue-level information of $s_j$, we employ a max-pooling operation to obtain the fusion information $f^{S_j}_t$ as follows:

$$ f^{S_j}_t[k] = \max \left\{ \left[ r^{S}_{1:t} [k], d^{S_j} [k] \right] \right\} $$

(7)

where $[.]_k$ represents the $k$-th value of a vector.

Dialogue State Distillation

In this module, we aim to learn to distill previous dialogue states from dialogue context without error propagation. To achieve this goal, we first use two BERTs (called the teacher network and the student network respectively) to encode different inputs: golden previous dialogue states and slot only states from dialogue context without error propagation. To implement this, we train the teacher network to tackle SOP and drive the student network to imitate it simultaneously. Thus, the student network can dynamically utilize relevant information of previous dialogue states without the gap between training and testing.

Base Encoder

For the teacher network, we concatenate all golden pairs $(s_j, v_{t-j-1}^t, 1 \leq j \leq J)$ of the $(t-1)$-th turn as the input for the $t$-th turn and use a BERT to encode it:

$$ SV_{t_j} = [\text{SLOT}_{t-j}^\text{tea}] \oplus s_j \oplus v_{t_j-1}^t, 1 \leq j \leq J $$

(8)

$$ H^\text{tea}_{t_j} = \text{BERT}_{\text{finetune}}([\text{CLS}] \oplus SV_{t_j} \oplus [\text{SEP}]) $$

(9)

where $[\text{SLOT}_{t-j}^\text{tea}]$ denotes special tokens and \(\cdot\) is a separation symbol. Based on $H_{t_j}^\text{tea}$, we use the representation at the position of $[\text{SLOT}_{t-j}^\text{tea}]$ to represent $(s_j, v_{t_j-1}^t)$, denoted as $h_{s_j,v_{t_j-1}}^\text{tea}$. For the student network, we use slot only concatenation as the input at any turn and use another BERT to encode it:

$$ S_{t_j} = [\text{SLOT}_{t-j}^\text{stu}] \oplus s_j, 1 \leq j \leq J $$

(10)

$$ H_{t_j}^\text{stu} = \text{BERT}_{\text{finetune}}([\text{CLS}] \oplus S_{t_j} \oplus [\text{SEP}]) $$

(11)

where $[\text{SLOT}_{t-j}^\text{stu}]$ denotes special tokens. Then, we use the representation at the position of $[\text{SLOT}_{t-j}^\text{stu}]$ to represent $s_j$ based on $H_{t_j}^\text{stu}$, denoted as $h_{s_j}^\text{stu}$. 

13836
Dialogue State Attention For the given dialogue context at the $t$-th turn and the slot $s_j$, we use $h_{j,t}^{tea}$ and $h_{j,t}^{stu}$ to retrieve the current dialogue context respectively by two different multi-head attention mechanisms as follows:

$$r_{j,t}^{tea} = \text{MultiHead}(h_{j,t}^{tea}, R_t, R_t) \tag{12}$$

$$r_{j,t}^{stu} = \text{MultiHead}(h_{j,t}^{stu}, R_t, R_t) \tag{13}$$

Distillation To push the teacher network to hold the slot information of the $t$-th turn, we train it to tackle State Operation Prediction (SOP). SOP aims to identify whether a slot updates its value compared with previous turns, while “0” is used if otherwise. We obtain its corresponding loss $\mathcal{L}_{sop}$ as follows:

$$p_{j,t}^{sop} = \sigma (W_1 \tanh (W_2 r_{j,t}^{tea})) \tag{14}$$

$$\mathcal{L}_{sop} = \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{J} - (y_{j,t}^{sop} \log p_{j,t}^{sop} + (1-y_{j,t}^{sop}) \log (1-p_{j,t}^{sop})) \tag{15}$$

where $\sigma$ denotes a sigmoid activation function, $y_{j,t}^{sop}$ denotes the golden label of slot $s_j$ for SOP, $W_1 \in \mathbb{R}^{1 \times d_{out}}$ and $W_2 \in \mathbb{R}^{d_{out} \times d_{out}}$ are trainable matrices. Note that $d_{out}$ is the dimension of $r_{j,t}^{tea}$.

Meanwhile, we use a mean square error loss (MSE) to calculate the similarity of the learned representations between the teacher and the student network for each slot. Thus, the student network can learn to distill previous dialogue state information from the teacher network by minimizing the loss. The distillation loss $\mathcal{L}_{\text{distill}}$ is obtained as follows:

$$\mathcal{L}_{\text{distill}} = \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{J} \text{MSE}(r_{j,t}^{tea}, r_{j,t}^{stu}) \tag{16}$$

After that, we formulate the overall loss $\mathcal{L}_{\text{total}}$ in the module as follows:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{sop} + (1-\alpha) \mathcal{L}_{\text{distill}} \tag{17}$$

where $\alpha (0 < \alpha < 1)$ is a balancing coefficient.

Inter-slot Contrastive Learning

In this module, we construct contrastive learning between slots to capture the slot co-update relations automatically. As our goal is to capture the co-update relations of slots, we only consider the slots whose labels of SOP are “1” in our contrastive learning. Assume that the SOP label of slot $s_j$ at the $t$-th turn (denoted as $s_j^t$) is “1” (i.e., $y_{j,t}^{sop} = 1$), we introduce the strategy of positive sample selection and negative sample selection, and the calculation of contrastive learning loss as follows:

Positive Sample Selection For the slot $s_j$ at the $t$-th turn, we select the slots whose labels of SOP are also “1” at the same turn as positive samples and the selected set $\mathcal{P} (s_j^t)$ is shown as follows:

$$\mathcal{P} (s_j^t) = \{ s_p^t : (y_{p,t}^{sop} = 1) \land (p \neq j) \} \tag{18}$$

As contrastive learning pulls representations of the given instance $s_j^t$ and the positive samples together, the similarity between the learned representations of slots which update in the same turn will be increased, thus helping the model capture the co-updated relations between different slots.

Negative Sample Selection For the slot $s_j$ at the $t$-th turn, we select negative samples from both turn-level and dialogue-level. For turn-level, we select the slots whose labels of SOP are “0” in the same turn as negative samples. And for dialogue-level, we select the slot $s_j$ at the other turns as negative samples. The selected contrastive learning set $\mathcal{N}_{\text{turn}} (s_j^t)$ is shown as follows:

$$\mathcal{N}_{\text{turn}} (s_j^t) = \{ s_n^t : (y_{n,t}^{sop} = 0) \land (n \neq j) \} \tag{19}$$

Figure 2: The architecture of our proposed model.
\[
N_{\text{dialogue}} (s_j^t) = \left\{ s_j^t : t' \neq t \right\}
\]
\[
I (s_j^t) = P (s_j^t) \cup N_{\text{turn}} (s_j^t) \cup N_{\text{dialogue}} (s_j^t)
\]
(20)
(21)
where \(N_{\text{turn}} (s_j^t)\) and \(N_{\text{dialogue}} (s_j^t)\) denote the turn-level and dialogue-level selected sets respectively. Contrastive learning pushes representations of the given instance \(s_j^t\) and the negative samples in \(N_{\text{turn}} (s_j^t)\) away. Thus, the similarity of the learned representations between the given instance \(s_j^t\) and the negative samples in \(N_{\text{turn}} (s_j^t)\) will be reduced. It helps capture the co-updated relations between slots at the same turn. On the other hand, we reduce the similarity of the learned representations between the same slot at different dialogue turns. As a result, distinguishing these subtle representation differences for the same slot can help our model perceive the state changes of each slot better as the dialogue continues.

**Contrastive Learning Loss** We first integrate the output of Encoder and Dialogue State Distillation, thus utilizing the previous dialogue state information:
\[
c_{t}^S = f_{t}^{S} + r_{t}^{Stu}
\]
(22)
Then, we follow (Chen et al. 2020b; Khosla et al. 2020) to map \(c_t^S\) into a new space, where contrastive learning is used to improve the quality of learning, as follows:
\[
z_t^{S} = W^3 \text{ReLu} \left(W^4 c_t^S \right)
\]
(23)
where \(W^3 \in \mathbb{R}^{d_1 \times d_2}\) and \(W^4 \in \mathbb{R}^{d_2 \times d_{out}}\) are trainable matrices. Note that \(d_1\) and \(d_2\) are the dimensional parameters. After that, we obtain the contrastive learning loss \(L_{\text{cl}}\) based on Normalized Temperature-scaled Cross Entropy (NT-Xent) (Chen et al. 2020b) as follows:
\[
L_{\text{cl}}^{j,t} = \sum_{s_j^t \in P (s_j^t)} \log \frac{\exp \left( \frac{\text{sim} (z_t^{S_j^t}, z_t^{S_p})}{\tau} \right)}{\sum_{s_j^t' \in I (s_j^t)} \exp \left( \frac{\text{sim} (z_t^{S_j^t}, z_t^{S_p'})}{\tau} \right)}
\]
\[
L_{\text{cl}} = \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{J} \frac{1}{|P (s_j^t)|} L_{\text{cl}}^{j,t}
\]
(24)
(25)
where \(\text{sim} (z_t^{S_j^t}, z_t^{S_p}) = z_t^{S_j^t \top} z_t^{S_p} / \|z_t^{S_j^t}\|_2 \|z_t^{S_p}\|_2\) and \(\tau\) is a temperature parameter.

**Decoder**
In this module, we aim to predict the value for each slot. Given the slot \(s_j\) at the \(t\)-th turn, the integrated representation \(c_t^{S_j}\) is first fed into a linear layer with layer normalization as follows:
\[
o_t^{S_j} = \text{LayerNorm} \left(W^5 c_t^{S_j} \right)
\]
(26)
where \(W^5 \in \mathbb{R}^{d_{out} \times d_{out}}\) is a trainable matrix.
After that, following (Ren et al. 2018; Wang et al. 2022b), we use L2 norm to obtain the distances between \(o_t^{S_j}\) and candidate values of \(s_j\). We obtain the loss \(L_{\text{value}}\) as follows:
\[
p_{j,t}^{\text{value}} = \frac{\exp \left( -\|o_t^{S_j} - h_{V_j}^{t} \|_2 \right)}{\sum_{v \in V(s_j)} \exp \left( -\|o_t^{S_j} - h_{V}^{j} \|_2 \right)}
\]
(27)
where \(h_{V_j}^{t}\) is the representation of golden value for \(s_j\) and \(V(s_j)\) denotes the candidate value set of \(s_j\).

**Optimization**
To push each module to focus on its optimization objective, we employ a two-phase optimization strategy. First, we optimize the joint objective of \(L_{\text{phase1}}\) as follows:
\[
L_{\text{phase1}} = L_{\text{value}} + L_{\text{dsd}}
\]
(29)
After that, we fix the parameters of the Dialogue State Distillation module based on development datasets and then optimize the joint objective of \(L_{\text{phase2}}\) as follows:
\[
L_{\text{phase2}} = L_{\text{value}} + L_{\text{cl}}
\]
(30)

**Performance Evaluation**

**Dataset** We evaluate our proposed model on the widely used MultiWOZ 2.0 (Budzianowski et al. 2018) and MultiWOZ 2.1 (Eric et al. 2020) datasets. MultiWOZ 2.0 includes over 10,000 dialogues and 35 slots across 7 domains. MultiWOZ 2.1 is the modified version of MultiWOZ 2.0 and corrects previous annotation errors of MultiWOZ 2.0.

**Metrics** Following previous works on the two datasets, we use Joint Goal Accuracy (Joint GA) as our evaluation metrics. It is calculated as the proportion of dialogue turns for which the value of each slot is correctly predicted.

**Training Details** For all modules, we follow (Wang et al. 2022b; Feng et al. 2022) to use the pre-trained BERT-base-uncased model for Base Encoder. The number of heads is set to 4 for the multi-head attention mechanism. In Encoder, the number of stacked layers in the transformer encoder is set to 6. In Dialogue State Distillation, we set the balancing coefficient \(\alpha\) to 0.8 and 0.6 for MultiWOZ 2.0 and MultiWOZ 2.1 respectively. In Inter-slot Contrastive Learning, we set \(d_1\) to 512, \(d_2\) to 512 and the temperature parameter \(\tau\) to 0.01. In the two-phase optimization strategy, the epoch of the first phase training is set to 100 and it stops early when the validation loss is not decreased for 15 consecutive epochs. As for the second phase training, we set the epoch to 15. Moreover, we set the batch size to 8 and 16 on four NVIDIA Tesla V100 GPUs during the first phase and second phase training respectively. Adam is used as the optimizer with learning rates of \(1 \times 10^{-4}\) and \(1 \times 10^{-5}\) during the first phase and second phase training respectively, and the warmup proportion is set to 0.1 during all training processes.

**Baselines** We use the following models as baselines for comparison with our proposed model: TripPy (Heck et al. 2020) combines three copy strategies for slot value predictions; DST-Picklist (Zhang et al. 2020a) first predicts slot updating information and then extracts the values of all slots via category classification; SST (Chen et al. 2020a) integrates schema graph and dialogue context, and controls the slot updating information from previous dialogue
### Performance Results

The experimental results in Joint GA are shown in Table 1. From the results, we observe that our model achieves the state-of-the-art performance on MultiWOZ 2.1 and almost the same performance as the state-of-the-art model LUNA on MultiWOZ 2.0. Further, we observe that previous baseline models cannot achieve the most competitive performance on both MultiWOZ 2.0 and MultiWOZ 2.1. Therefore, to compare the comprehensive performance among these models, we sum the performance on the two datasets for each model. As can be seen, our model outperforms the latest state-of-the-art model LUNA by 1.62%. Moreover, our model consistently achieves superior performance compared to previous dialogue states and our model uses knowledge distillation to tackle it.

Additionally, we compare the per-turn performance between STAR, LUNA and our model on MultiWOZ 2.1. From the results shown in Figure 3, we observe that our model is superior to LUNA and STAR in almost all dialogue turns. It further demonstrates the effectiveness of our model regardless of the length of dialogue.

### Ablation Study

We conduct an ablation study on the MultiWOZ 2.0 and MultiWOZ 2.1 datasets to analyze the effectiveness of each module in our model. As shown in Table 3, we observe that both the Dialogue State Distillation and Inter-slot Contrastive Learning modules contribute significantly to the performance of our proposed model regardless of the used dataset. Here, we focus the discussion on MultiWOZ 2.0. Firstly, if we remove the Dialogue State Distillation and Inter-slot Contrastive Learning modules separately, the performance of the proposed model will be dropped by 1.77% and 1.05% respectively in Joint GA. It demonstrates that both the Dialogue State Distillation and Inter-slot Contrastive Learning modules can boost the performance of the proposed model. Secondly, if we train with only $L_{value}$, the performance of the proposed model will be decreased significantly by 3.37% in Joint GA. It further demonstrates the effectiveness of distilling previous dialogue states from dialogue context and capturing the slot co-update relations.

### Further Analysis

In this section, we study the effectiveness of capturing the slot co-updated relations in our proposed model. First, we remove both the Dialogue State Distillation and Inter-slot Contrastive Learning modules, denoted as “Baseline”. Based on the “Baseline”, we use different SOP training losses, which include “Cross Entropy Loss” and “Contrastive Learning Loss−”, to replace the “Contrastive

---

Table 1: Experimental results in Joint GA (%) based on MultiWOZ 2.0 and MultiWOZ 2.1.

<table>
<thead>
<tr>
<th>Model</th>
<th>MultiWOZ 2.0</th>
<th>MultiWOZ 2.1</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>TripPy (2020)</td>
<td>53.51</td>
<td>55.32</td>
<td>108.83</td>
</tr>
<tr>
<td>DST-Picklist (2020a)</td>
<td>54.39</td>
<td>53.30</td>
<td>107.69</td>
</tr>
<tr>
<td>SST (2020a)</td>
<td>51.17</td>
<td>55.23</td>
<td>106.40</td>
</tr>
<tr>
<td>Transformer-DST (2020a)</td>
<td>54.64</td>
<td>55.35</td>
<td>109.99</td>
</tr>
<tr>
<td>Graph-DST (2020b)</td>
<td>52.78</td>
<td>53.85</td>
<td>106.63</td>
</tr>
<tr>
<td>FPDSC (turn-level) (2021)</td>
<td>55.03</td>
<td>57.88</td>
<td>112.91</td>
</tr>
<tr>
<td>FPDSC (dual-level) (2021)</td>
<td>53.17</td>
<td>59.07</td>
<td>112.24</td>
</tr>
<tr>
<td>STAR (2021)</td>
<td>54.53</td>
<td>56.36</td>
<td>110.89</td>
</tr>
<tr>
<td>DSGFNet (2022)</td>
<td>-</td>
<td>56.70</td>
<td>-</td>
</tr>
<tr>
<td>PPTOD (2022)</td>
<td>53.89</td>
<td>57.45</td>
<td>111.34</td>
</tr>
<tr>
<td>LUNA (2022b)</td>
<td><strong>55.31</strong></td>
<td>57.62</td>
<td>112.93</td>
</tr>
<tr>
<td>Our Model</td>
<td>55.30</td>
<td><strong>59.25</strong></td>
<td><strong>114.55</strong></td>
</tr>
</tbody>
</table>

Figure 3: Per-turn performance comparison in Joint GA (%) on MultiWOZ 2.1.
In this section, we discuss a dialogue snippet in the test set of the MultiWOZ 2.1 dataset. Table 2 shows the prediction results of DST by using “Our model w/o Distillation” and our model. “Our model w/o Distillation” means that the Dialogue State Distillation module is removed from our model and previous dialogue states are directly concatenated on the dialogue context in Equation (1). In the first turn of the dialogue snippet, the system mentions a hotel name “cambridge belfry” and the user does not show a clear attitude towards this hotel. Maybe the dialogue context given by the system is too salient, both models ignore the ambiguous attitude of the user and predict the value of “hotel-name” wrongly. At the second turn, “Our model w/o Distillation” utilizes the wrongly predicted information of “hotel-name” at the first turn and the information misleading the prediction of “Our model w/o Distillation”. As our model uses the Dialogue State Distillation module to distill previous dialogue state information without introducing error propagation, it avoids the disturbance of the wrongly predicted information and predicts the value of “hotel-name” as “none” correctly.

### Conclusion

In this paper, we propose a Dialogue State Distillation Network (DSDN) to utilize relevant information of previous dialogue states and fill the gap of utilization between training and testing, thus dynamically exploiting previous dialogue states without error propagation. In addition, an inter-slot contrastive learning loss is also proposed to effectively capture the slot co-update relations from dialogue context. Experimental results based on the MultiWOZ 2.0 and MultiWOZ 2.1 datasets have shown that our proposed model has achieved promising performance compared with state-of-the-art models for DST.
Acknowledgements
This work is supported by National Key Research and Development Program of China (Grant No. 2020YFC0832606), National Natural Science Foundation of China (Grant Nos. 61976021, U1811162), and Beijing Academy of Artificial Intelligence (BAAI).

References


