Correctable-DST: Mitigating Historical Context Mismatch between Training and Inference for Improved Dialogue State Tracking

Hongyan Xie[1], **Haoxiang Su**[1], **Shuangyong Song**[2], **Hao Huang**[1,4†]
Bo Zou[3], **Kun Deng**[3], **Jianghua Lin**[3], **Zhihui Zhang**[3] **and Xiaodong He**[3]
[1]School of Information Science and Engineering, Xinjiang University, Urumqi, China
[2]Department of Big Data and AI, China Telecom [3]JD AI Research, Beijing, China
[4]Xinjiang Provincial Key Laboratory of Multi-lingual Information Technology, Urumqi, China
{shiehyjy,haoxisu98,hwanghao}@gmail.com songshy@chinatelecom.cn
{cdzoubo,dengkun1,cdlinjianghua,cdzhangzhihui}@jd.com

Abstract

Recently proposed dialogue state tracking (DST) approaches predict the dialogue state of a target turn sequentially based on the previous dialogue state. During the training time, the ground-truth previous dialogue state is utilized as the historical context. However, only the previously predicted dialogue state can be used in inference. This discrepancy might lead to error propagation, i.e., mistakes made by the model in the current turn are likely to be carried over to the following turns. To solve this problem, we propose Correctable Dialogue State Tracking (Correctable-DST). Specifically, it consists of three stages: (1) a **Predictive State Simulator** is exploited to generate a previously "predicted" dialogue state based on the ground-truth previous dialogue state during training; (2) a **Slot Detector** is proposed to determine the slots with an incorrect value in the previously "predicted" state and the slots whose values are to be updated in the current turn; (3) a **State Generator** takes the name of the above-selected slots as a prompt to generate the current state. Empirical results show that our method achieves 67.51%, 68.24%, 70.30%, 71.38%, and 81.27% joint goal accuracy on MultiWOZ 2.0-2.4 datasets, respectively, and achieves a new state-of-the-art performance with significant improvements.

1 Introduction

Dialogue state tracking is the core module of a task-oriented dialogue system. It extracts the user's goal in each turn and represents the dialogue state as a set of (slot, value) pairs. Its performance will affect the decision prediction of the dialogue system.

Traditional neural network-based DST casts the prediction of slot values into a classification task(Mrkšić et al., 2017; Liu and Lane, 2017; Zhong et al., 2018; Ren et al., 2018; Nouri and Hosseini-Asl, 2018), requiring a predefined ontology which includes all candidate slot-value pairs. However, some undefined slot values (Xu and Hu, 2018) will appear in real scenarios. Therefore, current DST research focuses mainly on open vocabulary DST(Chao and Lane, 2019; Hosseini-Asl et al., 2020; Heck et al., 2020; Ham et al., 2020; Feng et al., 2021; Lin et al., 2021; Su et al., 2022), where the value of each slot is generated or extracted based on the dialogue history to resolve scalability and generalization issues of the predefined ontology-based approach, but these approaches often lack efficiency because they predict the dialogue state from scratch at every dialogue turn.

Some approaches utilize the dialogue state of the previous turn as a compact representation of the dialogue history and, based on that, generate the slot values to improve efficiency. Kim et al. (2020); Zeng and Nie (2020a) decompose the DST into two tasks: state operation prediction and value generation. But the performance of the state operation prediction will affect the performance of DST(Kim et al., 2020). Therefore, Chen et al. (2020); Lin et al. (2020); Yang et al. (2021); Tian et al. (2021) proposed to jointly model state operation prediction and value generation in an implicit way, and the dialogue state tracking is re-transformed into a single causal language model. Although these methods also work reasonably well, they suffer from error propagation (Zhao et al., 2021). Figure 1 illustrates the same errors that will appear in later dialogue turns, and dialogue state errors can be divided into three types (Quan and Xiong, 2020): (1) over prediction, which means the predicted dialogue state contains some redundant slot values. For example, the redundant slot value in the 5th turn state is "wandlebury country park"; (2) partial prediction, which means the predicted dialogue state lacks some slot values. The slot value "hotel" is missing in the state of the 1st turn; (3) erroneous prediction, which means the slot value of the pre-

[†] Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 876–889
December 7-11, 2022 ©2022 Association for Computational Linguistics

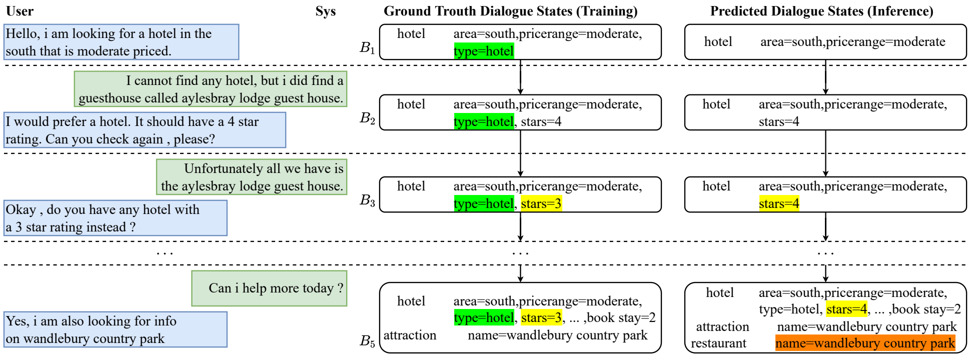
User
Sys
Ground Trouth Dialogue States (Training)
Predicted Dialogue States (Inference)
Hello, i am looking for a hotel in the south that is moderate priced.
B_1
hotel area=south,pricerange=moderate, type=hotel
hotel area=south,pricerange=moderate
I cannot find any hotel, but i did find a guesthouse called aylesbray lodge guest house.
I would prefer a hotel. It should have a 4 star rating. Can you check again , please?
B_2
hotel area=south,pricerange=moderate, type=hotel, stars=4
hotel area=south,pricerange=moderate, stars=4
Unfortunately all we have is the aylesbray lodge guest house.
Okay , do you have any hotel with a 3 star rating instead ?
B_3
hotel area=south,pricerange=moderate, type=hotel, stars=3
hotel area=south,pricerange=moderate, stars=4
...
Can i help more today ?
Yes, i am also looking for info on wandlebury country park
B_5
hotel area=south,pricerange=moderate, type=hotel, stars=3, ... ,book stay=2
attraction name=wandlebury country park
hotel area=south,pricerange=moderate, type=hotel, stars=4, ... ,book stay=2
attraction name=wandlebury country park
restaurant name=wandlebury country park

Figure 1: An example of dialogue state tracking is based on the recurrent state context representation. "User" and "Sys" represent user utterance and system response, respectively. Green indicates partial prediction, yellow indicates erroneous prediction, and orange indicates over prediction.

dicted state is not equal to that of the real dialogue state. In the 3rd turn, the value of hotel-stars is predicted to "4".

There are two factors contributing to the error propagation problem: (1) historical context mismatch between training and inference. Using the dialogue state of the previous turn as a compressed representation of the dialogue history, i.e., historical context, that is fed to the model is reasonable during training because the dialogue state of the previous turn is always correct. However, in the inference time, the previously predicted dialogue state may contain incorrect slot values; (2) these models cannot determine and correct errors in the previously predicted dialogue state. Because these models do not use the complete dialogue history as the input, they cannot compare the consistency of information between the previous dialogue history and the previously predicted dialogue state.

Based on the above, we propose a Correctable Dialogue State Tracking (Correctable-DST) model, as illustrated in Figure 2. We use the Predictive State Simulator to generate a pseudo "predicted" dialogue state, i.e., one that randomly inserts and deletes slot values based on the ground-truth dialogue state of the previous turn to alleviate the historical context mismatch. Then, the Slot Detector judges the slots with redundant value and missing value in the previously "predicted" dialogue state. Finally, when generating the current turn state, the name of the above slots is used as the prompt information fed to the State Generator, thus enhancing the ability of the model to correct the overprediction and partial prediction. In addition, the Slot Detector also predicts the slots whose value needs to be updated compared with the previous turn. Then it returns the slot's name to the State Generator to improve the model's update ability, thus reducing the error propagation caused by the erroneous prediction. Dialogue history is also used as an input to the model, which is a prerequisite for enabling the model to determine and correct errors in the previous dialogue state.

We evaluate the effectiveness of our model on MultiWOZ 2.0-2.4 datasets. Experimental results show that our model reaches 67.51%, 68.24%, 70.30%, 71.38% and 81.27% joint goal accuracy, outperforming previous strong baselines by **+10.58%**, **+7.51%**, **+9.81%**, **+5.51%** and **+6.64%**, respectively. Furthermore, a series of subsequent ablation studies were conducted to demonstrate the effectiveness of the proposed method. Our contributions are as follows:

(1) We propose a **Predictive State Simulator** to mitigate the historical context mismatch between training and inference by simulating the dialogue state predicted at the previous turn in the inference.

(2) We use the output slot by the **Slot Detector** as the prompt information when predicting the state of the current turn, which can not only help the model correct the errors in the previously "predicted" dialogue state but also reduce the current erroneous prediction.

(3) Experimental results on MultiWOZ 2.0-2.4 datasets show that our proposed Correctable-DST achieves new state-of-the-art performance.

2 Proposed Approach

Figure 2 illustrates the architecture of Correctable-DST, which includes Predictive State Simulator, Slot Detector, and State Generator. In this section, we elaborate on each module of this approach.

A dialogue can be represented as $X = \{(R_1, U_1), (R_2, U_2), \cdots, (R_T, U_T)\}$ with T turns

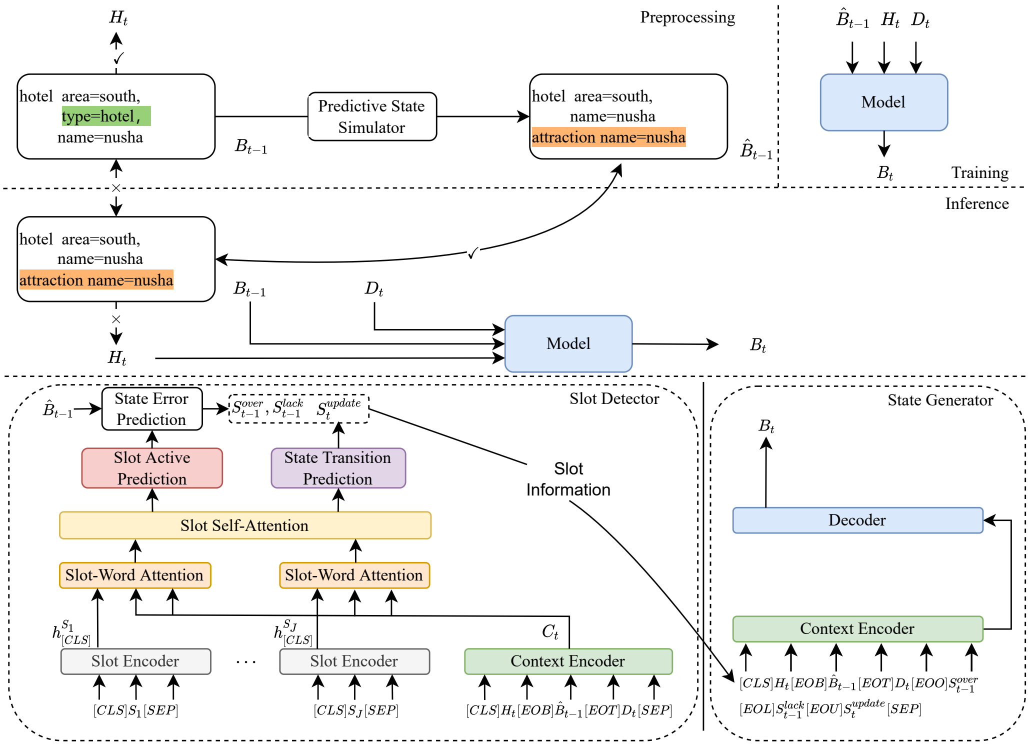

Figure 2: The overview of the proposed Correctable-DST. The model first generates previously "predicted" dialogue state $\hat{B}_{t-1}$ that may contain errors by Predictive State Simulator in training time, then Slot Detector obtains slot information S_{t-1}^{over}, S_{t-1}^{lack}, and S_{t-1}^{update}. Finally, the name of the above slot information is used as the prompt information fed to the State Generator.

where R_t represents system response and U_t represents user utterance at turn t. We define the dialogue state at turn t as $\mathcal{B}_t = \{(S_j, V_j^t) \mid 1 \leq j \leq J\}$ where V_j^t is the corresponding value of the slot S_j, and J represents the size of a set of predefined slots. Following (Ren et al., 2018), to represent domain and slot information, the term "slot" refers to the concatenation of domain and slot names, e.g., " <restaurant-name>".

2.1 Predictive State Simulator

The idea of the Predictive State Simulator is to generate previously "predicted" dialogue state $\hat{B}_{t-1}$ that may contain errors, as shown in the upper part of Figure 2.

We define two slot-level simulation strategies: (1) randomly deleting the slot value from the ground-truth previous dialogue state to simulate partial predictions. Specifically, if the slot value is not "none," the slot value is deleted with a probability of β. (2) Randomly insert a slot value to the ground-truth previous dialogue state to simulate over prediction. Specifically, if the slot value is "none", it will be replaced with the slot value that is not "none" in the ground-truth previous dialogue state with a probability of β. This tackle is inspired by our observation of SOM-DST (Kim et al., 2020) prediction results, where the redundant slot value is usually the value of other slots in the dialogue state.

2.2 Slot Detector

The slot detector is used to determine the slot with the incorrect value in the previously "predicted" dialogue state and the slot with the updated value of the current turn.

Slot Encoder The name of the slot $S_j (1 \leqslant j \leqslant J)$ is fed into BERT to generate a slot vector:

$$\boldsymbol{c}_{[\mathrm{CLS}]}^{S_j} = \mathrm{BERT}_{\mathrm{fixed}}\left([\mathrm{CLS}] \oplus S_j \oplus [\mathrm{SEP}]\right), \quad (1)$$

where $\mathrm{BERT}_{\mathrm{fixed}}$ represents the pre-trained BERT without fine tuning. $\boldsymbol{c}_{[\mathrm{CLS}]}^{S_j} \in \mathbb{R}^d$ is the output of [CLS] token from the slot S_j. d is the hidden size.

Context Encoder The full dialogue history and previously "predicted" dialogue state are used as a part of the context encoder input. By comparing whether the dialogue history information and the previously "predicted" dialogue state information are consistent or not, the errors in the previously "predicted" dialogue state are determined. The inputs to the context encoder consists of the dialogue utterances $D_t = R_t \oplus U_t$ at turn t, the dialogue history $H_t = D_1 \oplus D_2 \oplus \cdots \oplus D_{t-1}$ and the previously "predicted" dialogue state $\hat{B}_{t-1} = \bigoplus_{(S_j, V_j^{t-1}) \in \hat{\mathcal{B}}_{t-1}, V_j^{t-1} \neq none, 1 \leq j \leq J} S_j \oplus V_j^{t-1}$, where $\oplus$ is the operation of sequence concatenation. All the sub-sequences are concatenated with special segment tokens, i.e., $X_t = \text{[CLS]} \oplus H_t \oplus \text{[EOB]} \oplus \hat{B}_{t-1} \oplus \text{[EOT]} \oplus D_t \oplus \text{[SEP]}$, to calculate the context vector $\boldsymbol{C}_t$:

$$\boldsymbol{C}_t \quad = \quad \text{ContextEncoder}\left(X_t\right), \qquad (2)$$

where $\boldsymbol{C}_t \in \mathbb{R}^{d \times |X_t|}$ is the hidden states of the context encoder, $|X_t|$ is the input sequence length.

Slot Word Attention We employ a multi-head attention mechanism (Vaswani et al., 2017) to extract content specific to each slot $S_j (1 \leq j \leq J)$ based on the outputs of the context and slot encoders:

$$\boldsymbol{c}_t^{S_j} = \text{MultiHead}(\boldsymbol{c}_{\text{[CLS]}}^{S_j}, \boldsymbol{C}_t, \boldsymbol{C}_t), \qquad (3)$$

where $\boldsymbol{c}_t^{S_j} \in \mathbb{R}^d$ represents the slot-specific vector of j-th slot.

Slot Self-Attention We adopt a slot self-attention to learn the dependencies among the slots, which contains a stack of N identical layers, and each layer consists of a multi-head self-attention mechanism and a position-wise feed-forward network (FFN). Let $\tilde{\boldsymbol{C}}_1^S = [\boldsymbol{c}_t^{S_1}, \boldsymbol{c}_t^{S_2}, \ldots, \boldsymbol{c}_t^{S_J}] \in \mathbb{R}^{d \times J}$, the n-th layer's computations are:

$$\begin{aligned} \bar{\boldsymbol{C}}_n^S &= \text{MultiHead}(\tilde{\boldsymbol{C}}_{n-1}^S, \tilde{\boldsymbol{C}}_{n-1}^S, \tilde{\boldsymbol{C}}_{n-1}^S), && (4) \\ \tilde{\boldsymbol{C}}_n^S &= \text{FFN}(\text{ReLU}(\text{FFN}(\bar{\boldsymbol{C}}_n^S))). && (5) \end{aligned}$$

The final slot-specific vectors $\tilde{\boldsymbol{C}}_N^S$ are the outputs of the last layer, and $\tilde{\boldsymbol{C}}_N^S = [\boldsymbol{c}_t^{S_1}, \boldsymbol{c}_t^{S_2}, \ldots, \boldsymbol{c}_t^{S_J}] \in \mathbb{R}^{d \times J}$, where $\boldsymbol{c}_t^{S_j}$ is the final slot-specific vector of the j-th slot.

Auxiliary Classification Task We introduce two auxiliary binary classification tasks that are to be trained with DST model jointly: (1) slot activation prediction, to predict the activated slot in the previous turn; (2) state transition prediction, to predict whether the value of a slot is updated or not compared with the previous dialogue turn. Both tasks read the final slot-specific vectors $\tilde{\boldsymbol{C}}_N^S$ as the inputs. We define activation probability $P_{S_j^{t-1}}^{active}$ and transition probability $P_{S_j^t}^{update}$ as:

$$\begin{aligned} P_{S_j^{t-1}}^{active} &= \text{Sigmoid}(\boldsymbol{W}^{active}(\boldsymbol{c}_{N,t}^{S_j})^\top), && (6) \\ P_{S_j^t}^{update} &= \text{Sigmoid}(\boldsymbol{W}^{update}(\boldsymbol{c}_{N,t}^{S_j})^\top), && (7) \end{aligned}$$

where $\boldsymbol{W}^{active} \in \mathbb{R}^{1 \times d}$ and $\boldsymbol{W}^{update} \in \mathbb{R}^{1 \times d}$ are linear layers. $P_{S_j^{t-1}}^{active}$ represents the probability that the slot j was activated in the previous turn. $P_{S_j^t}^{update}$ is the probability that the slot needs to update the value in the current turn. The two probabilities are used to calculate an activation loss $\mathcal{L}_t^A$ and a transition loss $\mathcal{L}_t^U$:

$$\begin{aligned} \mathcal{L}_t^A = & - \frac{1}{J} \sum_{j=1}^{J} y_{t-1,j}^A \cdot \log P_{S_j^{t-1}}^{active} \\ & + (1 - y_{t-1,j}^A) \log P_{S_j^{t-1}}^{active}, && (8) \\ \mathcal{L}_t^U = & - \frac{1}{J} \sum_{j=1}^{J} y_{t,j}^U \cdot \log P_{S_j^t}^{update} \\ & + (1 - y_{t,j}^U) \log P_{S_j^t}^{update}, && (9) \end{aligned}$$

where $y_{t-1,j}^A$ and $y_{t,j}^U$ are 0-1 valued ground-truth activation label and transition label of the j-th slot, respectively.

Threshold-based Slot Detection The two probabilities in Equation (6) and (7) are then used to detect two slot sets: S_{t-1}^{active} and S_t^{update}. We define the slots activated in turn $t-1$ as $S_{t-1}^{active} = \{S_j^{t-1} \mid P_{S_j^{t-1}}^{active} \geqslant \alpha, 1 \leqslant j \leqslant J\}$, and the slots that updating value in turn t as $S_t^{update} = \{S_j^t \mid P_{S_j^t}^{update} \geqslant \alpha, 1 \leqslant j \leqslant J\}$. α is the threshold for classification. According to the slots activated S_{t-1}^{active}, the error in the previous dialogue state is indirectly determined as follows:

$$\begin{aligned} \hat{S}_{t-1} &= \{S_j^{t-1} \mid (S_j^{t-1}, V_j^{t-1}) \in \hat{\mathcal{B}}_{t-1}, \\ & \quad\ V_j^{t-1} \neq \text{none}, 1 \leqslant j \leqslant J\}, && (10) \\ S_{t-1}^{over} &= \hat{S}_{t-1} - \hat{S}_{t-1} \cap S_{t-1}^{active}, && (11) \\ S_{t-1}^{lack} &= S_{t-1}^{active} - \hat{S}_{t-1} \cap S_{t-1}^{active}, && (12) \end{aligned}$$

where $\hat{S}_{t-1}$ is the activated slots, S_{t-1}^{over} is the over-predicted slots and S_{t-1}^{lack} is the lack-predicted slots in the previously "predicted" dialogue state $\hat{\mathcal{B}}_{t-1}$. The three types of slot information S_{t-1}^{over}, S_{t-1}^{lack} and S_t^{update} will be used as prompt information when generating dialogue state.

2.3 State Generator

Figure 2 describes the State Generator with a general encoder-decoder architecture. The context encoder in the State Generator shares parameters with the context encoder in the Slot Detector. When generating the dialogue state B_t, the context encoder additionally encodes the over-predicted slots S_{t-1}^{over} and lack-predicted slots S_{t-1}^{lack} in the previously "predicted" dialogue state to help the model recover from earlier errors. Slots that require updated values S_t^{update} are encoded to reduce erroneous prediction in the current turn. Then, all the sub-sequences are concatenated with special segment tokens, i.e., $\hat{X}_t = \text{[CLS]} \oplus H_t \oplus \text{[EOB]} \oplus \hat{B}_{t-1} \oplus \text{[EOT]} \oplus D_t \oplus \text{[EOO]} \oplus S_{t-1}^{over} \oplus \text{[EOL]} \oplus S_{t-1}^{lack} \oplus \text{[EOU]} \oplus S_t^{update} \oplus \text{[SEP]}$, as the input to the context encoder:

$$\hat{\boldsymbol{C}}_t = \text{ContextEncoder}(\hat{X}_t), \qquad (13)$$

where $\hat{\boldsymbol{C}}_t \in \mathbb{R}^{d \times |\hat{X}_t|}$ represents the slot information augmented context vector, and $|\hat{X}_t|$ denotes the input sequence length. Then the decoder generates the user state:

$$\begin{aligned} \tilde{b}_t^l &= \text{Decoder}(B_t^{1:l-1} \mid \hat{\boldsymbol{C}}_t), & (14) \\ B_t &= \{b_t^l \mid b_t^l = \text{argmax}(P(b_t^l))\}, & (15) \\ P(b_t^l) &= W_{vocab} \cdot \tilde{b}_t^l, & (16) \end{aligned}$$

where $B_t^{1:l-1} = \{b_t^1, \cdots, b_t^{l-1}\}$, $l \in [1, |B_t|]$, and $|B_t|$ denotes the state sequence length. $W_{\text{vocab}} \in \mathbb{R}^{d \times N_{\text{vocab}}}$ is a linear layer projecting the hidden state feature space to N_{vocab} dimensional vocabulary space. The decoder gradually generates words over time until the end of generating [EOS], which is a special word that ends generation. Then it updates the dialogue state $\mathcal{B}_t$ by extracting the slot-value pairs from B_t. Equation 16 is used to calculate the cross entropy state loss:

$$\mathcal{L}_t^B = -\frac{1}{|B_t|} \sum_{l=1}^{|B_t|} \left(y_{t,l}^B\right)^\top \log P\left(b_t^l\right), \qquad (17)$$

where $y_{t,l}^B$ is the ground-truth token whose state needs to be generated at the l-th decoding step.

2.4 Optimization

We optimize the entire model parameters by jointly minimizing the sum of the three loss functions in Equation (8), (9) and (17)

$$\mathcal{L}_t = \mathcal{L}_t^A + \mathcal{L}_t^U + \mathcal{L}_t^B. \qquad (18)$$

3 Experiments

3.1 Datasets

The proposed model is evaluated on the MultiWOZ (Budzianowski et al., 2018) benchmark. Since there is annotation noise on MultiWOZ 2.0, some researchers have continuously revised the annotations and have released 4 variants of the dataset, namely MultiWOZ 2.1-2.4 (Eric et al., 2020; Zang et al., 2020; Han et al., 2021; Ye et al., 2021a). The processed dataset contains 5 domains, 17 slots, 30 (domain, slot) pairs, and over 4500 values.

3.2 Metrics

We use joint goal accuracy and final joint goal accuracy as our evaluation metrics. The joint goal accuracy is a measures the percentage of correct in all the dialogue turns. A turn is considered as correct only when all the values of slots are correctly predicted. The final joint goal accuracy is the proportion of examples (dialogues) where the predicted dialogue state of last turn exactly matches the ground-truth dialogue state of last turn.

3.3 Settings

We employ the BERT (Devlin et al., 2019) as the slot encoder to extract the slot vector, whose parameters are frozen during the training time. The BART (Lewis et al., 2020) was used as the state generator. For the slot-word attention and slot self-attention, we set the number of attention heads to 4. For the slot self-attention, we set the transformer layers to 6. We use the AdamW optimizer. The learning rate was set to 4e-5 for the BART and 1e-4 for the rests. The training batch size was set to 16 and the dropout (Srivastava et al., 2014) probability was set to 0.1. The threshold α was set to 0.5, and the β was set to 0.06. We report the mean results over multiple random seeds to reduce statistical errors. We use the data pre-processing script provided by Wu et al. (2019) for data preparation and data post-processing scripts are used to process the resulting sequence of dialogue states. We use the same hyperparameter configurations for all the experiments on MultiWOZ 2.0-2.4.

3.4 Baselines

We compare our approach with the following existing baselines, which are divided into two categories: (1) models that require a predefined ontology: *SST* (Chen et al., 2020) predicts dialogue states from dialogue utterances and schema graphs

	DH	PDS	MultiWOZ(%)				
			2.0	2.1	2.2	2.3	2.4
Predefined ontology							
DST-Picklist	Full History	×	54.39	53.30	-	-	-
SST	Previous Turn	✓	51.17	55.23	-	-	-
STAR	Full History	✓	$52.26^{\star}$/54.53	$54.08^{\star}$/56.36	**60.49***	**65.87***	**74.63***/$73.62^{\diamondsuit}$
DSS-DST	Full History	×	**56.93**	**60.73**	58.04	-	-
Open-vocabulary							
TRADE	Full History	×	48.62	45.60	$45.4^{\spadesuit}$	$49.2^{\clubsuit}$	$55.05^{\diamondsuit}$
TripPy	Full History	×	-	55.29	-	$63.0^{\clubsuit}$	$59.62^{\diamondsuit}$
SOM-DST	Previous Turn	✓	$52.61^{\star}$/51.38	$52.47^{\star}$/52.57	$53.27^{\star}$	$55.69^{\star}$/$55.5^{\clubsuit}$	$67.54^{\star}$/$66.78^{\diamondsuit}$
MinTL	Previous u Turns	✓	$51.41^{\star}$/52.1	$52.92^{\star}$/53.62	$55.82^{\star}$	$56.95^{\star}$	$67.56^{\star}$
TransformerDST	Previous Turn	✓	$53.71^{\star}$/54.64	$55.56^{\star}$/55.35	$55.64^{\star}$	$57.17^{\star}$	$69.74^{\star}$
SimpleTOD	Full History	×	-	55.76	$54.02^{\ddagger}$	$51.3^{\clubsuit}$	-
Seq2seq-DU	Full History	×	-	56.1	54.40	-	-
UBAR	Full History	✓	52.59	56.2	-	-	-
PPTOD	Full History	×	53.89	57.45	-	-	-
AGDST	No	✓	-	-	$57.26^{\ddagger}$	-	-
CorrectableDST	Full History	✓	**67.51** (±0.48)	**68.24** (±0.41)	**70.30** (±0.43)	**71.38** (±0.53)	**81.27** (±0.55)

Table 1: Joint goal accuracy (%) on the test set of MultiWOZ. DH: dialogue history. PDS: previous dialogue state. $\star$ means our reproduction results using a source code. $^{\spadesuit}$: the results borrowed from Zang et al. (2020). ‡: the results borrowed from Tian et al. (2021). $^{\clubsuit}$: results are cited from the 2.3 websites https://github.com/lexmen318/MultiWOz-coref. $^{\diamondsuit}$: the results borrowed from (Ye et al., 2021a). "-" indicates no public number is available. $\star$ means our reproduction results using a source code.

which contain slot relations in edges; *DS-Picklist* (Zhang et al., 2020) assumes a full ontology which is available and treats all domain-slot pairs as categorical slots; *STAR* (Ye et al., 2021b) propose a slot self-attention mechanism that can learn the slot correlations automatically. *DSS-DST* (Guo et al., 2021) propose a Dual Slot Selector which determines each slot whether to update the slot value or to inherit the slot value from the previous turn. (2) Models that can predict unseen values: *TRADE* (Wu et al., 2019) decodes the value for each slot using a copy-based GRU decoder; *TripPy* (Heck et al., 2020) makes use of three copy mechanisms to fill slots with values; *SOM-DST* (Kim et al., 2020) considers dialogue state as explicit fixed-sized memory and propose a selectively overwriting mechanism; *MinTL* (Lin et al., 2020) introduce Levenshtein belief spans, that allow efficient dialogue state tracking with a minimal generation length; *Transformer-DST* (Zeng and Nie, 2020a) propose a purely Transformer-based framework, where a single BERT works as both the encoder and the decoder; *SimpleTOD* (Hosseini-Asl et al., 2020) uses a single, causal language model trained on all sub-tasks recast as a single sequence prediction problem; *Seq2seq-DU* (Feng et al., 2021) formalizes DST as a sequence-to-sequence problem; *UBAR* (Yang et al., 2021) is acquired by fine-tuning the large pretrained unidirectional language model GPT-2 on the sequence of the entire dialogue session; *PPTOD* (Su et al., 2022) learn the primary TOD task completion skills from heterogeneous dialogue corpora. *AGDST* (Tian et al., 2021) learn more robust dialogue state tracking by amending the errors that exist in the primitive dialogue state.

3.5 Main Results

The results from our model and other baselines on the MultiWOZ 2.0-2.4 test sets are shown in Table 1. Our approach achieves state-of-the-art performance on these datasets with joint goal accuracy of 67.51%, 68.24%, 70.30%, 71.38%, and 81.27%. Compared to previous strong baselines, our approach achieves 10.58%, 7.51%, 9.81%, 5.51%, and 6.64% measurable performance promotion on MultiWOZ 2.0-2.4, respectively. Particularly, the joint goal accuracy on MultiWOZ 2.4 is beyond 80%. Similar (Kim et al., 2020; Lin et al., 2020; Zeng and Nie, 2020a; Ye et al., 2021b), thanks to MultiWOZ 2.4, which fines all the annotations in the validation set and test set on MultiWOZ 2.1, our model achieves higher joint goal accuracy on MultiWOZ 2.4 than on MultiWOZ 2.0-2.3.

As mentioned above, some approaches that predict the dialogue state of the target turn in sequence based on the previous dialogue state will suffer from the negative impact of error propagation. So we adopt a more stringent evaluation metric to measure the performance of the model, i.e., final joint goal accuracy. The final joint goal ac-

Model	MultiWOZ(%)				
	2.0	2.1	2.2	2.3	2.4
SOM-DST*	39.34	36.04	38.64	38.44	55.00
MinTL(BartLarge)*	38.70	39.04	41.04	40.24	56.96
Transformer-DST*	**39.69**	40.27	41.54	41.30	58.40
STAR*	38.93	**40.94**	**48.14**	**56.15**	**66.46**
Correctable-DST	**60.36**	**57.86**	**61.66**	**58.26**	**74.17**

Table 2: Final joint goal accuracy(%) on the MultiWOZ 2.0-2.4 test set. ⋆ means our reproduction results using the source code.

curacy of our model and other baselines on the test sets of MultiWOZ 2.0-2.4 is shown in Table 2. Our approach obtains 60.36% , 57.86%, 61.66%, 58.26% and 74.77% final joint goal accuracy, respectively, which has a significant improvement (20.67%, 16.92%, 13.52%, 2.11% and 7.71%) over the previous best results. The results demonstrate that our approach can effectively mitigate the error propagation problem.

3.6 Ablation Study

We evaluate the effectiveness of the proposed predictive state simulator and the additional encoding of slot information when generating dialogue state on the MultiWOZ 2.4, as shown in Table 3.

Model	**Joint Acc** (%)
basic model	77.38
+ Predictive State Simulator	79.15 (+1.77)
+ Slot Information	81.27 (+2.12)

Table 3: Ablation study of the Predictive State Simulator and the Slot Information on the MultiWOZ 2.4 test set in joint goal accuracy.

Our basic model is a multi-task model, which jointly trains the slot activation prediction, the state transition prediction and the dialogue state generation. The results show that our basic model also achieves good performance, attributed to the multi-task framework and the use of the BART. Still, we do not consider this to be our contribution. When we use the Predictive State Simulator, it will increase the joint goal accuracy by 1.77%. It has been proved that the Predictive State Simulator is used in training to simulate the predicted dialogue state in inference so that the model can adapt to the noise in inference. Based on using the Predictive State Simulator, we encode additional slot information to generate a dialogue state, and the results increased by 2.12%. This demonstrates the importance of slot information as prompt information when generating the current turn dialogue state.

3.6.1 Ablation on Predictive State Simulator

As aforementioned, we consider that over-prediction and partial-prediction make the historical context mismatched between training and inference, which eventually leads to the problem of error propagation. To verify this, we evaluate the mentioned random insertion and deletion strategies to generate a previously "predicted" dialogue state. As shown in Table 4, we observe that randomly inserting a slot value to the ground-truth previous dialogue state results in a 1.14% improvement, and randomly deleting a slot value from the ground truth previous dialogue state shows a gain of 0.58%. When we use both, the joint goal accuracy is increased by 1.77%.

Model	**Joint Acc**(%)
basic model	77.38
+ insert	78.52 (+1.14)
+ delete	77.96 (+0.58)
+ both	79.15 (+1.77)

Table 4: The ablation study of the two approaches of predictive state simulator on the MultiWOZ2.4 test set with joint goal accuracy.

Table 5 shows the performance of the model using different simulation probability β. We can see that, as expected, the performance of the model starts to grow when β is increased. However, the performance of the model decreases slightly when β is further increased. We believe that increasing β to a certain extent can increase the training samples' diversity and improve the model's robustness. On the other hand, the appropriate β can minimize the historical context difference between the training and reasoning processes.

β	**Joint Acc**(%)
0	77.38
0.03	78.58
0.06	**81.27**
0.09	80.25
0.12	79.30

Table 5: Simulation probability analysis on the MultiWOZ2.4 test set with joint goal accuracy.

3.6.2 Ablation on Slot Information

When generating a dialogue state, we use three types of slot information as a part of the input sequence. We conduct an ablation study to explore the effectiveness of different slot information, as shown in Table 6. The results show that the joint goal accuracy of the model is increased by 1.93%

when the over-predicted slots S^{over}_{t-1} is fed to the model. When the under-predicted slots S^{lack}_{t-1} is fed to the model, the joint goal accuracy of the model is increased by 0.33%. We guess that the slot information S^{over}_{t-1} and S^{lack}_{t-1} can help the model correctly the errors in the previously predicted dialogue state because they point out the wrong slot in the previously predicted dialogue state. Similarly, the slots value update information S^{update}_{t} is fed to the model, the joint goal accuracy of the model is increased by 1.47%, which means it can help the model update the dialogue state to mitigate historical context mismatches caused by erroneous prediction in the future.

Model	**Joint Acc**(%)
basic model + predictive state simulator	79.15
+ over-predicted slots	81.08
+ lack-predicted slots	79.48
+ slots that updating value	80.62
+ above all	81.27

Table 6: The ablation study of three slot information on the MultiWOZ2.4 with joint goal accuracy.

3.7 Error Analysis

In this section, the errors made by the model on MultiWOZ 2.4 are analyzed. We counted the proportion of the three error types for the dialogue state. About 65% of the errors were due to partial prediction, 13% were due to over prediction, and 22% were due to erroneous prediction. Out of these, 40% of partial prediction, 62% of over prediction, and 33% of erroneous prediction will eventually be corrected by our model. Note that there are no annotation errors in the MultiWOZ2.4 test set. Statistics show that although we do not design a particular scheme to correct erroneous prediction, it also shows comparable performance. Then we further observed the experimental results and found that the common mistakes in the generative model, e.g. **express by holiday inn cambridge** is predicted to **cafe by holiday inn cambridge**, accounted for more than 90% of the erroneous prediction, and only the common mistakes in the generative model were corrected.

4 Related Work

Traditional DST models formulate DST as a value classification task for each slot, assuming all values are available (Liu and Lane, 2017; Zhong et al., 2018; Nouri and Hosseini-Asl, 2018). In practice, this is a limiting assumption because there are a large number of possible values in real life, and it is impractical to enumerate all possible values. Therefore, the current research work mainly focuses on the open vocabulary DST(Chao and Lane, 2019; Hosseini-Asl et al., 2020; Heck et al., 2020; Ham et al., 2020; Feng et al., 2021; Lin et al., 2021; Su et al., 2022), where the value of each slot is generated or extracted based on the dialogue history. Kim et al. (2020) handle this problem by decomposing DST into two sub-tasks: state operation prediction and value generation. At each turn, whether the value in the previous dialogue state is modified or not or how to modify the value is determined by the discrete operations predicted by the state operation. Many recent works (Zeng and Nie, 2020a,b) follow this approach. Besides, some works (Chen et al., 2020; Lin et al., 2020; Yang et al., 2021) put forward jointly model state operation prediction and value generation in an implicit way, which means the current turn of dialogue and the previous dialogue state are used as input sequences, and the dialogue state tracking is re-transformed into a single causal language model. In addition, some task-oriented pre-training models have been proposed, Hosseini-Asl et al. (2020) used a single, causal language model trained on all sub-tasks recast as a single sequence prediction problem. Su et al. (2022) introduced a new dialogue multi-task pre-training strategy that allows the model to learn the primary TOD task completion skills from heterogeneous dialogue corpora. Feng et al. (2021) formalizes DST as a sequence-to-sequence problem. Yang et al. (2021) is acquired by fine-tuning the large pretrained unidirectional language model GPT-2 on the sequence of the entire Dialogue session.

Due to some methods (Kim et al., 2020; Zeng and Nie, 2020a; Chen et al., 2020; Lin et al., 2020; Yang et al., 2021) suffer from error propagation (Zhao et al., 2021). Tian et al. (2021) propose an amendable generation method which improves the output dialogue state through a two-pass decoding process, this is similar to automatic post-editing methods in the field of machine translation(do Carmo et al., 2021), there has first been a decoder producing the primitive output, to which it adds a second decoder, creating an improved version of that output. But the DST model improved by "two-pass decoding" may still produce wrong output, and the error will be propagated when predicting the subsequent states.

5 Conclusion

In this paper, we propose a new correctable dialogue state tracking approach. This approach generate the previous "predicted" dialogue state during training through the Predictive State Simulator. Then, the Slot Detector outputs the slot information. Finally, the State Decoder will correct the errors in the previously "predicted" dialogue state according to the slot information and reduce the current erroneous prediction to alleviate the error propagation problem. Experimental results show that our model achieves the state-of-the-art performance of 67.51%, 68.24%, 70.30%, 71.38%, and 81.27% achieving significant improvements (10.58%, 7.51%, 9.81%, 5.51%, and 6.64%) joint goal accuracy over the previous best results on the MultiWOZ 2.0-2.4.

Limitations

The first limitation of our approach is that the model is less efficient. As mentioned above, we need to judge and correct the errors in the previously predicted dialogue state depending on the dialogue history, which inevitably leads to too long a historical context for the input model, and the slot detector and the state generator need to encode the historical context.

The second limitation is that our model cannot detect erroneous predictions. We have tried simulating erroneous prediction with a predictive state simulator, and then designing a classification task in the slot detector to detect slots where the real slot value is not equal to the predicted slot value, but this did not have an advantage. We conjecture that simple classification tasks may fail to detect such finer-grained value-level errors.

Acknowledgement

We would like to thank the anonymous reviewers for their useful feedback. This work was supported by the Opening Project of Key Laboratory of Xinjiang, China (2020D04047), Natural Science Foundation of China (61663044), and the National Key R&D Program of China (2020AAA0107902).

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Guan-Lin Chao and Ian Lane. 2019. BERT-DST: Scalable End-to-End Dialogue State Tracking with Bidirectional Encoder Representations from Transformer. In *Proc. Interspeech 2019*, pages 1468–1472.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Félix do Carmo, Dimitar Shterionov, Joss Moorkens, Joachim Wagner, Murhaf Hossari, Eric Paquin, Dag Schmidtke, Declan Groves, and Andy Way. 2021. A review of the state-of-the-art in automatic post-editing. *Machine Translation*, 35(2):101–143.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 422–428.

Yue Feng, Yang Wang, and Hang Li. 2021. A sequence-to-sequence approach to dialogue state tracking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1714–1725.

Jinyu Guo, Kai Shuang, Jijie Li, and Zihan Wang. 2021. Dual slot selector via local reliability verification for dialogue state tracking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 139–151.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592.

Ting Han, Ximing Liu, Ryuichi Takanabu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. 2021. Multiwoz 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation. In

CCF International Conference on Natural Language Processing and Chinese Computing, pages 206–218. Springer.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Weizhe Lin, Bo-Hsiang Tseng, and Bill Byrne. 2021. Knowledge-aware graph-enhanced gpt-2 for dialogue state tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7871–7881.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. Mintl: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405.

Bing Liu and Ian Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. *Proc. Interspeech*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788.

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking model. *arXiv preprint arXiv:1812.00899*.

Jun Quan and Deyi Xiong. 2020. Modeling long context for task-oriented dialogue state generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7119–7124.

Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. 2018. Towards universal dialogue state tracking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2780–2786.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-task pre-training for plug-and-play task-oriented dialogue system.

Xin Tian, Liankai Huang, Yingzhan Lin, Siqi Bao, Huang He, Yunyi Yang, Hua Wu, Fan Wang, and Shuqi Sun. 2021. Amendable generation for dialogue state tracking. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 80–92.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1457.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14230–14238.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2021a. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. *arXiv preprint arXiv:2104.00773*.

Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021b. Slot self-attentive dialogue state tracking. In *Proceedings of the Web Conference 2021*, pages 1598–1608.

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. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117.

Yan Zeng and Jian-Yun Nie. 2020a. Jointly optimizing state operation prediction and value generation for dialogue state tracking. *arXiv preprint arXiv:2010.14061*.

Yan Zeng and Jian-Yun Nie. 2020b. Multi-domain dialogue state tracking based on state graph. *arXiv preprint arXiv:2010.11137*.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, S Yu Philip, Richard Socher, and Caiming Xiong. 2020. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*.

Jeffrey Zhao, Mahdis Mahdieh, Ye Zhang, Yuan Cao, and Yonghui Wu. 2021. Effective sequence-to-sequence dialogue state tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7486–7493.

V. Zhong, C. Xiong, and R. Socher. 2018. Global-locally self-attentive dialogue state tracker. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

A Data Statistics

The MultiWOZ dataset includes 8,438 multi-turn dialogues in training set with an average of 13.5 turns per dialogue. For the test and validation set, each includes 1,000 multi-turn dialogues with an average of 14.7 turns per dialogue. The average number of domains per dialogue is 1.8 for training, validation, and test sets. In fact, the MultiWOZ dataset includes a total of 35 (domain, slot) pairs across 7 domains. However, Only 5 domains (restaurant, hotel, attraction, taxi, train) are used in our experiment because the other 2 domains (hospital, police) only appear in the training set. The statistics of dialogue in these 5 domains are shown in Table 7.

	attraction	hotel	restaurant	taxi	train
slot	area name type	area bookday bookpeople bookstay internet name parking pricerange stars type	area bookday bookpeople booktime food name pricerange	arriveby departure destination leaveat	arriveby bookpeople day departure destination leaveat
train	3,381	3,103	2,717	3,813	1,654
val	416	484	401	438	207
test	394	494	395	437	195

Table 7: The dataset statistics of MultiWOZ 2.0-2.4.

B Accuracy per Slot on MultiWOZ 2.4 Test Set

Table 8 shows the accuracy per slot on MultiWOZ 2.4 test set.

C Sample Prediction Output

Tables 9 and 10 shows the prediction output in all turns for 2 example dialogues: MUL0842 and SNG1026.

Domain-Slot	Our Model
attraction-area	98.71
attraction-name	97.98
attraction-type	99.04
hotel-area	98.59
hotel-book day	99.70
hotel-book people	99.62
hotel-book stay	99.78
hotel-internet	98.64
hotel-name	97.07
hotel-parking	98.62
hotel-price range	99.01
hotel-stars	99.17
hotel-type	98.72
restaurant-area	99.42
restaurant-book day	99.82
restaurant-book people	99.70
restaurant-book time	99.66
restaurant-food	98.78
restaurant-name	98.28
restaurant-price range	99.42
taxi-arrive by	100
taxi-departure	99.16
taxi-destination	99.55
taxi-leave at	100
train-arrive by	100
train-book people	99.89
train-day	99.33
train-departure	99.73
train-destination	99.95
train-leave at	100

Table 8: The detailed results of accuracy (%) per slot on MultiWOZ 2.4 test set.

Turn 1	User:	can you help me find a place to eat ? i am looking for somewhere in the east with a moderate price .
	Ground-truth Dialogue State:	(restaurant-pricerange, moderate),(restaurant-area, east)
	Prcdictcd Dialogue Statc:	(restaurant-pricerange, moderate),(restaurant-area, east)
Turn 2	System:	there is the curry prince and rajmahal restuarants serving indian food . then there is the pizza hut fen ditton serving italian food . which would you like more information on ?
	User:	i think i would like to try rajmahal . could you book me a table for 1 on monday at 16:15 ?
	Ground-truth Dialogue State:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east),
	Prcdictcd Dialogue Statc:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east),
Turn 3	System:	your reservation has been booked . your reference number is q9drq55j . please be aware that your table will only be reserved until 16:30 . can i provide any more assistance ?
	User:	i am also looking for things to do in the same general area as the restaurant .
	Ground-truth Dialogue State:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east), (attraction-area, east)
	Predicted Dialogue State:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east), (attraction-area, east)
Turn 4	System;	sure , are you interested in museums , sports , parks , entertainment , swimming or boating ?
	User:	are there any cinemas ? if there is not what about something that has to do with multiple sports ?
	Ground-truth Dialogue State:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east), (attraction-type, cinema|multiple sports),(attraction-area, east)
	Predicted Dialogue State:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east), (attraction-type, multiple),(attraction-area, east)
Turn 5	System:	there are not any cinemas in the general area , but what types of sports are you interested in ?
	User:	all of them . what s there ?
	Ground-truth Dialogue State:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east), (attraction-type, multiple sports),(attraction-area, east)
	Predicted Dialogue State:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east), (attraction-type, multiple sports),(attraction-area, east)
Turn 6	System:	the cherry hinton village centre is the only 1 located there , their address is colville road , cherry hinton . anything else i can help you with ?
	User:	can i please get the phone number
	Ground-truth Dialogue State:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east), (attraction-type, multiple sports),(attraction-name, cherry hinton village centre), (attraction-area, east)
	Predicted Dialogue State:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east), (attraction-type, multiple sports),(attraction-name, cherry hinton village centre), (attraction-area, east)
Turn 7	System:	yes , the phone number is 01223576412 . how else can i help ?
	User:	that will be all , thank you very much for all of your help .
	Ground-truth Dialogue State:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east), (attraction-type, multiple sports),(attraction-name, cherry hinton village centre), (attraction-area, east)
	Predicted Dialogue State:	(restaurant-book day, monday),(restaurant-book people, 1),(restaurant-book time, 16:15), (restaurant-pricerange, moderate),(restaurant-name, rajmahal),(restaurant-area, east), (attraction-type, multiple sports),(attraction-name, cherry hinton village centre), (attraction-area, east)

Table 9: Full set of predicted dialogue states for dialogue ID MUL0842 in MultiWOZ2.4.

Turn 1	User:	can you help me find a moderate price place to stay that include -s free parking ?
	Ground-truth Dialogue State:	(hotel-parking, yes),(hotel-pricerange, moderate)
	Prcdictcd Dialogue Statc:	(hotel-parking, yes),(hotel-pricerange, moderate)
Turn 2	System:	acorn guest house has free parking and is moderate -ly priced . would you like more information ?
	User:	does it also have free wifi and does it have 2 stars ?
	Ground-truth Dialogue State:	(hotel-name, acorn guest house),(hotel-parking, yes),(hotel-pricerange, moderate), (hotel-stars, 2),(hotel-internet, yes)
	Prcdictcd Dialogue Statc:	(hotel-name, acorn guest house),(hotel-parking, yes),(hotel-pricerange, moderate), (hotel-stars, 2)
Turn 3	System:	no it does not . but the ashley hotel has 2 stars and free parking . it is a moderate also .
	User:	is there somewhere with both free wifi and parking ?
	Ground-truth Dialogue State:	(hotel-parking, yes),(hotel-pricerange, moderate),(hotel-stars, 2),(hotel-internet, yes)
	Predicted Dialogue State:	(hotel-name, yes),(hotel-pricerange, moderate),(hotel-stars, 2),(hotel-internet, yes)
Turn 4	System;	yes . the ashley hotel has both free wifi and parking . would you like to make a reservation ?
	User:	i am not quite ready to make a reservation yet , but could you please get the address for me ?
	Ground-truth Dialogue State:	(hotel-name, ashley hotel),(hotel-parking, yes),(hotel-pricerange, moderate),(hotel-stars, 2), (hotel-internet, yes)
	Predicted Dialogue State:	(hotel-name, ashley hotel),(hotel-parking, yes),(hotel-pricerange, moderate),(hotel-stars, 2), (hotel-internet, yes)
Turn 5	System:	the ashley hotel is located at 74 chesterton road . would you like any other information about it ?
	User:	yes , do you have their postcode please ?
	Ground-truth Dialogue State:	(hotel-name, ashley hotel),(hotel-parking, yes),(hotel-pricerange, moderate),(hotel-stars, 2), (hotel-internet, yes)
	Predicted Dialogue State:	(hotel-name, ashley hotel),(hotel-parking, yes),(hotel-pricerange, moderate),(hotel-stars, 2), (hotel-internet, yes)
Turn 6	System:	the postcode is cb41er . can i help you with anything else ?
	User:	what part of town is the ashley located ?
	Ground-truth Dialogue State:	(hotel-name, ashley hotel),(hotel-parking, yes),(hotel-pricerange, moderate),(hotel-stars, 2), (hotel-internet, yes)
	Predicted Dialogue State:	(hotel-name, ashley hotel),(hotel-parking, yes),(hotel-pricerange, moderate),(hotel-stars, 2), (hotel-internet, yes)
Turn 7	System:	they are located in the north .
	User:	perfect , thank you . that is all i need . bye .
	Ground-truth Dialogue State:	(hotel-name, ashley hotel),(hotel-parking, yes),(hotel-pricerange, moderate),(hotel-stars, 2), (hotel-internet, yes)
	Predicted Dialogue State:	(hotel-name, ashley hotel),(hotel-parking, yes),(hotel-pricerange, moderate),(hotel-stars, 2), (hotel-internet, yes)

Table 10: Full set of predicted dialogue states for dialogue ID SNG1026 in MultiWOZ2.4.