electronics

MDPI

Article

RSP-DST: Revisable State Prediction for Dialogue State Tracking

Qianyu Li [1], Wensheng Zhang [1,2,*], Mengxing Huang [1,*], Siling Feng [1] and Yuanyuan Wu [1]

[1] School of Information and Communication Engineering, Hainan University, Haikou 570100, China
[2] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
* Correspondence: wensheng.zhang@ia.ac.cn (W.Z.); huangmx09@hainanu.edu.cn (M.H.)

Abstract: Task-oriented dialogue systems depend on dialogue state tracking to keep track of the intentions of users in the course of conversations. Although recent models in dialogue state tracking exhibit good performance, the errors in predicting the value of each slot at the current dialogue turn of these models are easily carried over to the next turn, and unlikely to be revised in the next turn, resulting in error propagation. In this paper, we propose a revisable state prediction for dialogue state tracking, which constructs a two-stage slot value prediction process composed of an original prediction and a revising prediction. The original prediction process jointly models the previous dialogue state and dialogue context to predict the original dialogue state of the current dialogue turn. Then, in order to avoid the errors existing in the original dialogue state continuing to the next dialogue turn, a revising prediction process utilizes the dialogue context to revise errors, alleviating the error propagation. Experiments are conducted on MultiWOZ 2.0, MultiWOZ 2.1, and MultiWOZ 2.4 and results indicate that our model outperforms previous state-of-the-art works, achieving new state-of-the-art performances with 56.35, 58.09, and 75.65% joint goal accuracy, respectively, which has a significant improvement (2.15, 1.73, and 2.03%) over the previous best results.

Keywords: revisable state prediction; dialogue state tracking; error propagation; joint goal accuracy; task-oriented dialogue systems

Citation: Li, Q.; Zhang, W.; Huang, M.; Feng, S.; Wu, Y. RSP-DST: Revisable State Prediction for Dialogue State Tracking. *Electronics* **2023**, *12*, 1494. https://doi.org/10.3390/electronics12061494

Academic Editor: George A. Tsihrintzis

Received: 21 February 2023
Revised: 17 March 2023
Accepted: 20 March 2023
Published: 22 March 2023

1. Introduction

Based on the great improvement of natural language human–computer interaction technology, task-oriented dialogue systems (Google Assistant, Tmall Genie, and Apple Siri et al.) are playing the part in ticket booking, restaurant reservations, and other practical scenarios. A classical pipelined task-oriented dialogue system involves four kernel modules: natural language understanding (NLU), dialogue state tracking (DST), dialogue policy learning (DPL), and natural language generation (NLG) [1–3]. DST is a critical task aiming at keeping track of the intents of users at each conversation turn and representing them in the forms of a group of (*slot*, *value*) pairs, i.e., dialogue state [4,5]. Additionally, the information in the state is utilized to determine the next system actions of DPL and NLG [3,6]. Figure 1 shows an example of dialogue ID PMUL2279 from the dataset, dialogue states extracted from conversations are a group of (*slot*, *value*) pairs, such as (*attraction-name*, *corpus christi*), (*restaurant-area*, *centre*), (*restaurant-food*, *African*), and so on.

Over the past few years, a great number of approaches about DST have been proposed, making significant improvements [3,7,8]. Traditional DST methods predict the value of every slot on the basis of a predefined ontology which is composed of all candidate (*slot*, *value*) pairs [4,9,10]. They encoder the dialogue history or current utterance, and then score all potential (*slot*, *value*) pairs of which the highest scoring value is the predicted value for the slot. Based on the advantage of exploiting the previous dialogue state as a compact representation of the previous dialogue history, some approaches model the previous dialogue state and dialogue context jointly when predicting the current dialogue state, achieving good performance [11,12]. Recently, approaches based on open vocabulary have been proposed to address the problem of unseen slot values. They divide DST into

two sub-tasks: state operation prediction and value generation [13–16]. The results of the state operation prediction at each turn determine whether the state of the previous dialogue turn should be revised or not. Though previous state-of-the-art (SOTA) DST approaches exhibit satisfactory performance, we observed that the errors in the prediction of the value of the slots in the current dialogue turn of these models are easily carried over to the next turn, and unlikely to be revised in the next turn, resulting in error propagation. Actually, these models perform a one-time slot value prediction process, lacking a double-checking process to detect and revise the errors of the current dialogue turn. The absence of such a revision process can result in some potential errors not being identified and revised. As shown in Figure 1, some existing models predict the value of slot *restaurant-pricerange* is *none* at the fourth dialogue turn, while the ground truth label is *expensive*. Because of lacking a double-checking process, the wrong dialogue state is continued to the next turn, leading to error propagation.

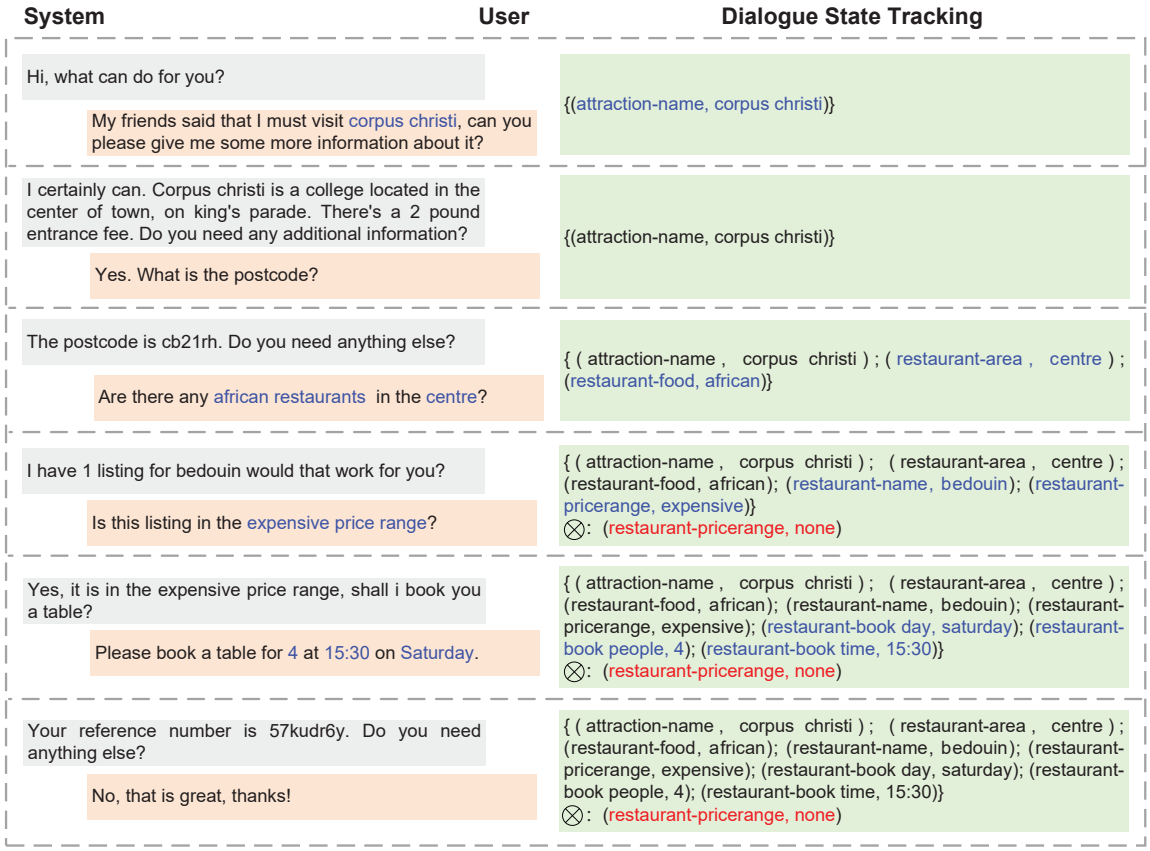


Figure 1. An example of dialogue state tracking. Each turn consists of a system response (grey) and a user utterance (orange). The blue colour denotes the new state appearing at that turn. The dialogue state tracker (green) tracks all the (*slot*, *value*) pairs until the current turn. "⊗" represents the incorrect result marked with red colour which is predicted by some existing methods.

To address the above problem, we propose the revisable state prediction for dialogue state tracking (RSP-DST), which constructs a two-stage slot value prediction process consisting of an original prediction and a revising prediction. Specifically, the first stage of RSP-DST jointly models the previous dialogue state and dialogue context to predict original dialogue states, then the second stage leverages the dialogue context to revise the original dialogue state. The second stage plays the role of reviser which is expected to detect and revise the errors existing in the original dialogue state. With the help of such a two-stage prediction process, RSP-DST is unlikely to carry erroneous dialogue states over to the next turn, alleviating the error propagation. Figure 2 represents an example of a two-stage dialogue state prediction process of RSP-DST. In this example, the revising prediction

process detects the error of slot *restaurant-pricerange* existing in the original dialogue state and revises it with the right value.

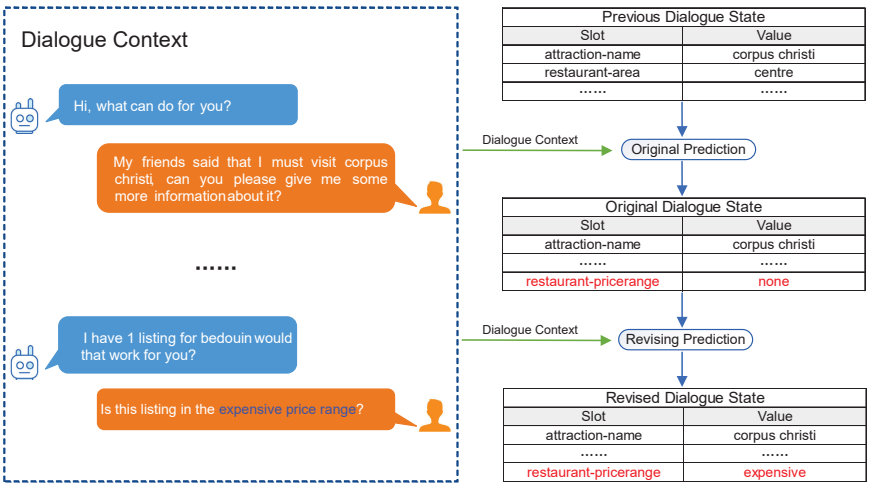


Figure 2. An example of a two-stage dialogue state prediction process of RSP-DST. The user wants to book an expensive restaurant by "Is this listing in the expensive price range". In the two-stage dialogue state prediction process, the mistake of slot *restaurant-pricerange* existing in the original dialogue state is revised with the right value in the revising prediction process.

Comprehensive experiments are conducted on three benchmark datasets: MultiWOZ 2.0 [17], MultiWOZ 2.1 [18], and MultiWOZ 2.4 [19]. Results show that RSP-DST consistently outperforms all prior works and achieves a new SOTA performance with joint goal accuracy of 56.35, 58.09, and 75.65%, respectively, which has a significant improvement (1.82, 1.73, and 2.03%, respectively) on the top of the previous best model. Additionally, we provide a series of ablation studies to demonstrate the effectiveness of the revising prediction.

The rest organization of this paper is as follows. We introduce the research background and related work in Section 2. In Section 3, we introduce our method RSP-DST in detail. Section 4 provides some information about experiments, including datasets, evaluation metrics, baselines, and other implementation details during the training process. In Section 5, we report experimental results on three benchmark datasets and analyse the results in detail. Finally, we present our conclusions and future works in Section 6.

2. Related Work

Traditional statistical DST models combine semantic features extracted by the NLU module or jointly learn speech understanding to predict the current dialogue states [20–24]. However, these methods tend to be highly dependent on complex domain-specific lexicons and hand-crafted features, making them difficult to scale to new dialogue domains.

With the development of deep learning-based dialogue systems, deep neural networks such as CNN, RNN, LSTM, GRU, and so on attract researchers to apply it in DST [3,7–9,14,21,25–29]. The performance of these models achieves obvious improvement over the previous statistical DST models. Owing to the success of the pre-trained language models such as BERT [30] and GPT-2 [31] in the field of natural language processing (NLP), recent studies in DST focus on building a new model on the basis of these pre-trained language models and achieve good performance [10,12,13,15,32–37]. DST in those approaches is considered to be a classification or generation problem.

Classification methods usually consist of an encoder and a classifier. The encoder outputs the representation of the conversation context, and the classifier scores all potential values which come from the predefined ontology. The highest scoring value is chosen to be predicted to the slot. Based on the advantage of exploiting the previous dialogue state as a compact representation of the previous dialogue history, some approaches model

the previous dialogue state and dialogue context jointly when predicting the current dialogue state, achieving good performance. SUMBT [10] applies BERT [30] and a slot-word attention mechanism to learn the relationships between dialogue context and slots. CHAN [32] enhances the interaction between dialogue history and slots by constructing a hierarchical attention network. DST-picklist [33] constructs a reading comprehension framework to match the values of categorical and non-categorical slots from ontologies. STAR [12] extracts slot-specific information with a slot self-attention mechanism and achieves a good performance.

Most generation methods are based on open vocabulary and do better in handling the problem of unseen slot values and domains. They divide DST into two sub-tasks: state operation prediction and value generation. Specifically, the state operation prediction task plays the role of encoding dialogue context and previous dialogue state and outputting the result of the state operation, then the value generation task predicts the value of each slot based on the result of the state operation prediction [13–15]. TripPy [13] utilizes three copy mechanisms to fill slots and the values of slots are obtained from the dialogue context. TRADE [14] is made up of three modules: dialogue context encoder, slot gate, and value generator. This method encodes utterances with GRU and predicts the value of slots from the dialogue context with a copy mechanism. SOM-DST [15] consists of a state operation predictor and a slot value generator. State operation predictor jointly encodes the last two turns of conversation and the previous dialogue state with a BERT and outputs the state operations on each slot. The slot value generator predicts the value of the slot of which the result of state operation is UPDATE. SimpleTOD [34] formulates DST as a single causal language model to generate system response, system action, and dialogue state. It encodes the dialogue context with GPT-2. Seq2Seq-DU [29] is a sequence-to-sequence approach that handles the unseen domain problem by applying schema descriptions. It encodes utterance and schema with BERT and generates dialogue state with an LSTM state decoder. SAF [35] constructs a self-supervised attention flow framework composed of DRS (dialogue response selection) and DST to learn dependencies among the dialogue and domain/slot. SPSF-DST [36] proposes a stack-propagation framework and a slot-masked attention mechanism to enhance the performance of DST.

Moreover, advances in technologies, such as reading comprehension [38], knowledge graph [39–42], graph attention network [11,43–46], and reinforcement learning [47,48], have led researchers to see more potential in DST research. Some studies formulate the DST task as a reading comprehension task and achieve a good performance [24,49,50]. DSTQA [42] formulates the DST task as a question-answering problem and learns the correlations among (*domain*, *slot*) pairs with a dynamically evolving knowledge graph. SST [44] is a classification method, it constructs a schema graph and then fuses information from dialogue utterances and the schema graph with a graph attention network.

All the aforementioned DST approaches perform a one-time slot value prediction process and are unlikely to detect and revise the errors of the current dialogue turn. Furthermore, errors are easily carried over to the next turn, resulting in error propagation. The proposed RSP-DST model constructs a two-stage dialogue state prediction process, which can detect and revise errors existing in the current turn, alleviating the error propagation problem.

3. Methods

In this section, we introduce the proposed method RSP-DST in the following aspects: problem definition, the original dialogue state prediction, the revising dialogue state prediction, and the training objective. Figure 3 represents the architecture of RSP-DST. RSP-DST-base outputs the original dialogue state, removing the revising prediction process. The symbols used in this section are listed and described in Table A1.

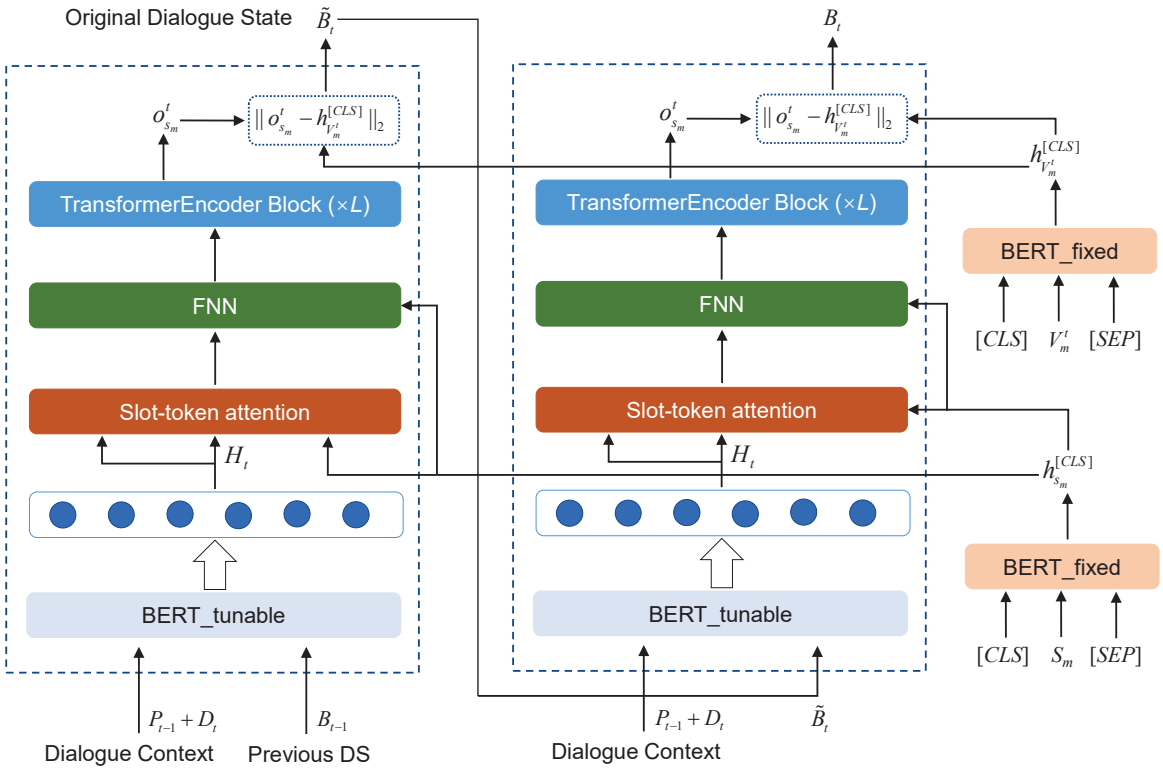


Figure 3. The overview of RSP-DST. BERT_tunable indicates that we will fine-tune the parameters of the BERT-base in the process of training. D_t denotes the current dialogue context composed of the system response and the user utterance, and P_{t-1} represents dialogue history. The input sequence at turn t is $[CLS] \oplus P_{t-1} \oplus B_{t-1} \oplus [SEP] \oplus D_t \oplus [SEP]$. BERT_fixed is utilized to encode values and slots, freezing the parameters in the training phase.

3.1. Problem Definition

Suppose that $\mathcal{D} = \{D_1, D_2, D_3, \ldots, D_T\}$ represents a dialogue with T turns where $D_t = (R_t, U_t)$, R_t symbolizes the system response and U_t symbolizes the user utterance at turn t, respectively. Furthermore, $\mathcal{S} = \{S_1, S_2, S_3, \ldots, S_M\}$ is a predefined slot set where M denotes the total number of slots in all dialogue domains. $B_t = \{(S_m, V_m^t) \mid 1 \leq m \leq M\}$ represents the dialogue states at turn t, where $S_m \in \mathcal{S}$ is the m-th slot and V_m^t is the corresponding value of S_m. Following previous works [10,12,51], the slot S_m is made up of the domain and slot names connected with a special token (i.e., $<$ *domain-slot* $>$) to include both domain and slot information. For instance, "*hotel-stars*" represents the slot "*stars*" in the "*hotel*" domain rather than "*stars*".

Consequently, in the task of DST, our work is to train a dialogue state tracker $\mathcal{T} : \mathcal{D} \rightarrow B_t$ that makes full use of the dialogue context $\mathcal{D}$ to predict the dialogue state B_t as accurately as possible at turn t [12].

3.2. Original Dialogue State Prediction

3.2.1. Dialogue Context Encoder

Following previous works [10,12,15], the pre-trained language model BERT-base [30] is applied to encode the dialogue context to obtain semantic vector representations.

As introduced before, $D_t = R_t \oplus U_t$ denotes the dialogue utterance at turn t, where $\oplus$ denotes the operation of sequence concatenation. Furthermore, $P_t = D_1 \oplus D_2 \oplus \ldots \oplus D_t$ represents the dialogue history until t. To a certain degree, B_{t-1} can be seen as a compact representation of the dialogue history [15]. Considering this, we treat the B_{t-1} as part of the input as well in the training phase. $B_t = [B_1^t; B_2^t; \ldots; B_M^t]$ where ; is a special token playing the part of discriminating different dialogue states and $B_m^t = [S_m, V_m^t]$ in which only non-NONE slots are considered.

Consequently, the entire input sequence of dialogue context at turn t can be represented as:

$$X_t = [CLS] \oplus P_{t-1} \oplus B_{t-1} \oplus [SEP] \oplus D_t \oplus [SEP], \tag{1}$$

where $[CLS]$ is a special token and usually has a role in aggregating all token-specific representations of the sequence, and $[SEP]$ is an auxiliary token for separation and also plays the part in marking the end of the sentence. Hence, feeding X_t to BERT, we have:

$$\boldsymbol{H}_t = BERT_{fine\text{-}tune}(X_t), \tag{2}$$

where $\boldsymbol{H}_t \in \mathbb{R}^{|X_t| \times d}$, $|X_t|$ and d denote the number of total tokens in X_t and the BERT output dimension, respectively. Note that $BERT_{fine\text{-}tune}$ indicates that we will fine-tune the parameters of the BERT-base in the process of training.

3.2.2. Slot and Value Encoder

Following previous works [10,32], the fixed BERT-base is leveraged to encode slot S_m and its corresponding value V_m^t. Note that we will freeze the parameters of BERT-base in the course of training which is different from the dialogue context encoder. Additionally, $[CLS]$ is an auxiliary special token for aggregating all token-specific representations of the sequence. In terms of slots and their values, we adopt the aggregated representation of the whole input sequence which is represented by the vector representation of $[CLS]$. As follows:

$$\boldsymbol{h}_{s_m}^{[CLS]} = BERT_{fixed}([CLS] \oplus S_m \oplus [SEP]), \tag{3}$$

$$\boldsymbol{h}_{V_m^t}^{[CLS]} = BERT_{fixed}([CLS] \oplus V_m^t \oplus [SEP]). \tag{4}$$

3.2.3. Slot Attention

We employ a multi-head attention mechanism [52] to extract slot-relevant information from the dialogue context. We treat the slot representation $\boldsymbol{h}_{s_m}^{[CLS]}$ as query vector. Furthermore, the key and value vectors are represented as $\boldsymbol{H}_t$. We have:

$$\boldsymbol{h}_{s_m,t} = MultiHead\left(\boldsymbol{h}_{s_m}^{[CLS]}, \boldsymbol{H}_t, \boldsymbol{H}_t\right), \tag{5}$$

where $\boldsymbol{h}_{s_m t} \in \mathbb{R}^d$. Then, we concatenate the $\boldsymbol{h}_{s_m,t}$ and $\boldsymbol{h}_{s_m}^{[CLS]}$ to retain the information of slot name and transform the merged vector with a feedforward neural network (FFN) which has two fully connected layers with a ReLU activation function in between, as follows:

$$\tilde{\boldsymbol{h}}_{s_m,t} = FFN\left(Concat\left(\boldsymbol{h}_{s_m,t}, \boldsymbol{h}_{s_m}^{[CLS]}\right)\right), \tag{6}$$

where $\tilde{\boldsymbol{h}}_{s_m,t} \in \mathbb{R}^d$. Following this operation, we obtain the complete token-aware slot representations $\tilde{\boldsymbol{H}}_{s,t} = \left[\tilde{\boldsymbol{h}}_{s_1,t}, \tilde{\boldsymbol{h}}_{s_2,t}, \tilde{\boldsymbol{h}}_{s_3,t}, \ldots, \tilde{\boldsymbol{h}}_{s_M,t}\right]$, while each slot representation in $\tilde{\boldsymbol{H}}_{s,t}$ does not fully share information. For this reason, we adopt a transformer encoder [52] to learn the correlation among slots. There are L identical layers in the transformer encoder module and each of which has two sub-layers. Specifically, the self-attention layer is the first sub-layer, which is used to obtain interacted information, and the feedforward neural network (FFN) is the second sub-layer. Formally, we have:

$$\hat{\boldsymbol{H}}_{s,t} = TransformerEncoder\left(\tilde{\boldsymbol{H}}_{s,t}\right), \tag{7}$$

where $\hat{\boldsymbol{H}}_{s,t} = \left[\hat{\boldsymbol{h}}_{s_1 t}, \hat{\boldsymbol{h}}_{s_2,t}, \hat{\boldsymbol{h}}_{s_3,t}, \ldots, \hat{\boldsymbol{h}}_{s_M,t}\right]$ represents the mutual interaction information, and $\hat{\boldsymbol{h}}_{s_m,t}$ means the slot-related representation of the slot S_m at turn t which is expected to be the closest to the semantic vector representation of the true value of the slot S_m. In light of the output of the BERT-base is normalized by layer normalization [53], we feed $\hat{\boldsymbol{h}}_{s_m,t}$

to a normalization layer preceded by a linear transformation layer. The process can be formulated as:

$$\boldsymbol{o}_{s_m}^t = LayerNorm(Linear(Dropout(\hat{\boldsymbol{h}}_{s_m,t}))). \tag{8}$$

3.2.4. Value Prediction

In this part, we predict the value of slot S_m according to $\boldsymbol{o}_{s_m}^t$ and the semantic vector representation of value $V'_m \in V_m$, i.e., $\boldsymbol{h}_{V'_m}^{[CLS]}$, where V_m denotes the value space of the slot S_m. Firstly, we calculate the distance between $\boldsymbol{o}_{s_m}^t$ and $\boldsymbol{h}_{V'_m}^{[CLS]}$. After this, the prediction of slot S_m is determined according to the distance of the candidate value, choosing the smallest one. In accordance with [27], our distance metric is the L2-norm. In this way, the probability distribution of the value prediction can be formulated as follows:

$$p\left(V_m^t \mid P_{t-1}, D_t, B_{t-1}, S_m\right) = \frac{\exp\left(-\left\|\boldsymbol{o}_{s_m}^t - \boldsymbol{h}_{V_m^t}^{[CLS]}\right\|_2\right)}{\sum_{V'_m \in V_m} \exp\left(-\left\|\boldsymbol{o}_{s_m}^t - \boldsymbol{h}_{V'_m}^{[CLS]}\right\|_2\right)}. \tag{9}$$

Based on the value prediction probability distribution, we obtain the dialogue state $\tilde{B}_t$ at turn t which is the original dialogue state. Then, the complete process of original dialogue state prediction can be formulated as:

$$\tilde{B}_t = OriginalState(P_{t-1}, D_t, B_{t-1}). \tag{10}$$

3.3. Revising Dialogue State Prediction

In order to revise the potential errors in $\tilde{B}_t$ at turn t, we utilize a revising dialogue state prediction module which takes the original dialogue state and the dialogue context as input and obtains the revised state. The complete process of revising dialogue state prediction process can be formulated as:

$$B_t = RevisedState\left(P_{t-1}, D_t, \tilde{B}_t\right). \tag{11}$$

where B_t is the revised dialogue state, the new input sequence of the revising dialogue state prediction module is made up of the original dialogue state and context. The revising prediction module shares the same parameters with the original prediction module.

3.4. Training Objective

In the original state prediction module, the training objective is the negative log-likelihood loss, we have:

$$\mathcal{L}_{dist\_original} = \sum_{m=1}^{M} -\log\left(p\left(V_m^t \mid P_{t-1}, D_t, B_{t-1}, S_m\right)\right). \tag{12}$$

Similar to the original state prediction module, the loss of the revising dialogue state prediction module is also the negative log-likelihood. Thus, we have:

$$\mathcal{L}_{dist\_revised} = \sum_{m=1}^{M} -\log\left(p\left(V_m^t \mid P_{t-1}, D_t, \tilde{B}_t, S_m\right)\right). \tag{13}$$

The total loss of model RSP-DST is to minimize the sum of the above two losses:

$$\mathcal{L}_{dist} = \mathcal{L}_{dist\_original} + \mathcal{L}_{dist\_revised} \tag{14}$$

4. Experiments

4.1. Datasets

We assess the performance of RSP-DST on three progressive datasets: MultiWOZ 2.0 [17], MultiWOZ 2.1 [18] and MultiWOZ 2.4 [19]. MultiWOZ 2.0 (https://www.repository.cam.ac.uk/bitstream/handle/1810/280608/MULTIWOZ2.zip?sequence=3&isAllowed=y, accessed on 23 April 2022) is a publicly available and large-scale task-oriented dialogue dataset in multiple domains, which contains more than 10,000 dialogues and spans seven distinct domains. MultiWOZ 2.1 (https://www.repository.cam.ac.uk/bitstream/handle/1810/294507/MULTIWOZ2.1.zip?sequence=1&isAllowed=y, accessed on 23 April 2022) and MultiWOZ 2.4 (https://github.com/smartyfh/MultiWOZ2.4/blob/main/data/MULTIWOZ2.4.zip, accessed on 12 August 2022) are revised versions of MultiWOZ 2.0. Specifically, MultiWOZ 2.1 corrected approximately 32% of the dialogue state annotation errors in MultiWOZ 2.0. On the basis of MultiWOZ 2.1, MultiWOZ 2.4 kept the training set unchanged and manually corrected dialogue state annotation errors in verification and test sets.

Following previous works [10,11,14,15,32], five domains (*train, taxi, hotel, attraction, restaurant*) will be utilized in our experiments, while the other two domains (*police, hospital*) will not be used because they appear infrequently in the training set and not contained in the test and validation sets. The resulting datasets include 30 *domain-slot* pairs and 17 different slots in five domains. Table 1 reports the data statistics in detail. The data pre-processing procedures we used on MultiWOZ 2.0, 2.1, and 2.4 are similar to [14].

Table 1. Data statistics of MultiWOZ 2.1. Three columns on the right summarize the total number of dialogues in every domain.

Domain	Slots	Train	Valid	Test
Train	arriveby, leaveat, day, book people, destination, departure	3103	484	494
Taxi	arriveby, leaveat, destination, departure	1654	207	195
Hotel	area, parking, type, stars, book people, book day, book stay, pricerange, name, internet	3381	416	394
Attraction	area, type, name	2717	401	395
Restaurant	area, book people, book time, book day, name, pricerange, food	3813	438	437

4.2. Evaluation Metric

The evaluation metric we adopt in our model is joint goal accuracy (JGA) [14]. Joint goal accuracy is the ratio of dialogue turns of which slots have been filled with correct values according to the ground truth. In particular, the ground truth value will be set to none if its slot is not presented in a turn. Furthermore, note that we also need to predict all none slots. The joint goal accuracy is 1.0 if, and only if, all slots are correctly predicted at every turn, otherwise it is 0. As well as joint goal accuracy, we also calculate slot accuracy and various other evaluation metrics for a more detailed and comprehensive analysis of the RSP-DST model.

4.3. Baselines

We make a comparison with the following approaches.

TRADE: TRADE [14] is made up of an utterance encoder, a slot-gate, and a dialogue state generator. The method generates a dialogue state from the input utilizing a copy mechanism, handling the cross-domain phenomenon.

SOM-DST: SOM-DST [15] is an open vocabulary-based approach. The method treats the dialogue state as an explicit fixed-sized memory and overwrites the memory selectively at each turn.

SimpleTOD: SimpleTOD [34] is an end-to-end model based on GPT-2, which generates system responses, system actions, and dialogue states by changing sub-tasks in task-oriented dialogue tasks into a single causal language model.

TripPy: TripPy [13] utilizes three copy mechanisms to fill slots and the values of slots are obtained from the dialogue context.

Seq2Seq-DU: Seq2Seq-DU [29] is a sequence-to-sequence approach that handles the unseen domain problem by applying schema descriptions.

SAF: SAF [35] constructs a self-supervised attention flow framework composed of DRS (dialogue response selection) and DST to learn dependencies among the dialogue and domain/slot.

SPSF-DST: SPSF-DST [36] proposes a stack-propagation framework and a slot-masked attention mechanism to enhance the performance of DST.

CHAN: CHAN [32] proposes a hierarchical attention network to enhance the interaction between dialogue history and slot. The approach also applies an adaptive objective to alleviate the slot imbalance problem.

SST: SST [44] constructs a schema graph and then fuses information from dialogue utterances and the schema graph with a graph attention network (GAT) [43].

DST-picklist: DST-picklist [33] constructs a reading comprehension framework to match the values of categorical and non-categorical slots from ontologies.

STAR: STAR [12] is an ontology-based approach, which employs a slot self-attention module to automatically learn the correlation among slots.

4.4. Implementation Details

We employ the BERT-base-uncased (https://huggingface.co/bert-base-uncased, accessed on 23 April 2022) model as encoders of RSP-DST, of which the weights of the dialogue context encoder need to be fine-tuned and the weights of the slot names encoder and slot values encoder are fixed. The BERT-base has 12 layers with 768 hidden units and 12 self-attention heads. For multi-head attention in our experiments, we set the head counts to four and the hidden size as 768. The transformer encoder module has six layers (i.e., L). During the training process, the AdamW [54] optimizer is adopted and the warm-up proportion is set to 0.05. We utilize different learning rates in regard to different parts of our model. Specifically, the peak learning rates of the dialogue context encoder in RSP-DST are set to 4×10^{-5} and other parts of the model are 1×10^{-4} since the dialogue context encoder is a pre-trained BERT model which does not need to be trained from scratch. We set the training batch size to 16 and the dropout [55] rate to 0.1. The maximum length of the input sequence is 512. Additionally, the word dropout [56] method is employed in our approach by randomly replacing the dialogue utterance tokens with a special token [UNK] with a rate of 0.1. We apply the same hyperparameter settings on MultiWOZ 2.0, MultiWOZ 2.1, and MultiWOZ 2.4. All experiments were performed on one NVIDIA Tesla V100 32G card.

5. Results and Discussion

5.1. Main Results

Table 2 shows the performance of our RSP-DST model in comparison to various baselines. As reported in Table 2, our model RSP-DST consistently outperforms all baselines on both joint goal accuracy and slot accuracy, achieving a new SOTA performance. RSP-DST-base removes the revising dialogue state prediction module and achieves 54.42, 56.31, and 74.64% joint goal accuracy, respectively. With the help of the revising prediction process, RSP-DST achieves 56.35, 58.09 and 75.65% joint goal accuracy, respectively, with a great improvement $(1.93, 1.78,$ and $1.01\%, respectively)$ on the top of RSP-DST-base. The results of RSP-DST also have a significant improvement $(2.15, 1.73$ and $2.03\%, respectively)$ on the top of the previous best model, STAR [12]. Meanwhile, RSP-DST also achieves 97.54, 97.75 and 98.95% slot accuracy, respectively. The outstanding performance illustrates the effectiveness of RSP-DST in dialogue state tracking tasks.

Table 2. Joint goal accuracy of RSP-DST and baselines on test sets of MultiWOZ 2.0, MultiWOZ 2.1, and MultiWOZ 2.4.

Model	Joint Goal Accuracy (%)			Slot Accuracy (%)		
	MWZ2.0	**MWZ2.1**	**MWZ2.4**	**MWZ2.0**	**MWZ2.1**	**MWZ2.4**
TRADE [14]	48.62	45.60	55.05	96.92	96.55	97.62
SOM-DST [15]	51.72	53.01	66.78	-	97.15	98.38
SimpleTOD [34]	51.37	51.89	-	-	-	-
TripPy [13]	53.51	55.18	59.62	-	97.48	97.94
Seq2Seq-DU [29]	-	56.10	-	-	-	-
SAF [35]	-	51.60	-	-	97.50	-
SPSF-DST [36]	54.88	54.32	-	-	-	-
CHAN [32]	53.06	53.38	68.25	-	97.39	98.52
SST [44]	51.17	55.23	-	-	-	-
DST-picklist [33]	54.39	53.30	-	-	97.40	-
STAR [12]	54.20 †	56.36	73.62	97.33 †	97.59	98.85
RSP-DST-base *	54.42	56.31	74.64	97.44	97.62	98.87
RSP-DST	**56.35**	**58.09**	**75.65**	**97.54**	**97.75**	**98.95**

† The results reproduced using the source code and other results reported in the literatures. * RSP-DST-base outputs original dialogue state, removing the revising prediction module.

In addition, we represent the performance of RSP-DST in the single- and multi-domains in Figure 4. Conversation scenarios in practice usually involve multiple domains. DST encounters the challenging phenomenon of domain migration in the multi-domain conversation scenario. As shown in Figure 1, the conversation involves two domains (*attraction*, *restaurant*). In Figure 4, we compare our model RSP-DST with STAR [12], TripPy [13], and SOM-DST [15]. STAR is a classification approach and is the previous best baseline model. TripPy and SOM-DST are generation approaches based on open vocabulary. TripPy utilizes three copy mechanisms to fill slots and the slot values are obtained from the dialogue context. SOM-DST treats the dialogue state as an explicit fixed-sized memory and overwrites the memory selectively at each turn. Both of them achieve great improvements in DST, and they are often used as baseline models for comparison in various performance experiments in recent DST studies. From Figure 4, it is obvious that the RSP-DST model achieves better performance in comparison to STAR, TripPy, and SOM-DST, in single- and multi-domain conversation scenarios. RSP-DST-base achieves 68.28% in the single-domain and 54.32% in the multi-domain. Furthermore, with the help of the revising prediction process, the joint goal accuracy reaches 70.66% in the single-domain and 55.99% in the multi-domain, with improvements of 2.38% and 1.67%, respectively.

5.2. Domain-Specific Joint Goal Accuracy and Per-Slot Accuracy

In this section, we further explore the performance of RSP-DST in domain-specific and each slot. We provide the results of domain-specific accuracy in Figure 5 where we compare RSP-DST with STAR [12], TripPy [13], and SOM-DST [15]. The accuracy of domain-specific is measured on a subset of the predicted dialogue state containing all slots belonging to the specific domain. Furthermore, for each domain, only the dialogues that are domain-active are considered. From Figure 5, it is clear that RSP-DST consistently outperforms the comparison approaches, achieving a joint goal accuracy 77.11% in the *attraction* domain, 60.36% in the *hotel* domain, 74.07% in the *restaurant* domain, 67.35% in the *taxi* domain, and 80.47% in the *train* domain.

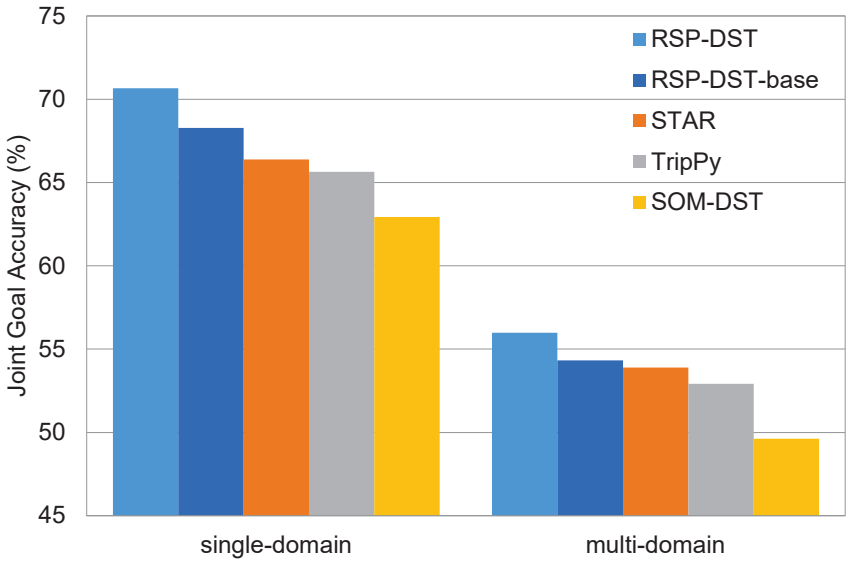


Figure 4. Joint goal accuracy of the single- and multi-domain on the test set of MultiWOZ 2.1.

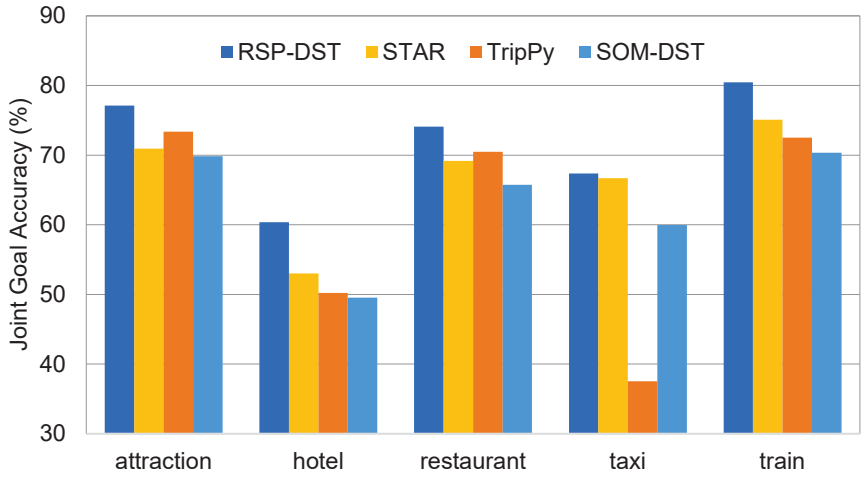


Figure 5. Domain-specific joint goal accuracy on the test set of MultiWOZ 2.1.

Figure 6 represents the per-slot accuracy of RSP-DST and TripPy [13]. The dialogues of the domain to which the slot belongs are used to calculate the per-slot accuracy. As shown in Figure 6, RSP-DST outperforms TripPy [13] in most slots, while we notice that RSP-DST is inferior to TripPy [13] for the "*attraction-name*", "*hotel-name*", and "*restaurant-name*" slots which are associated with entity names. In terms of these slots, their values in practical applications are so multifarious that they cannot be completely predefined. In addition, these values are usually directly informed by the user during the conversation. TripPy [13] is an open vocabulary-based method that generates values of slots from the dialogue context with three copy mechanisms, which is likely to be more efficient to predict these values. This observation encourages us to enhance the extraction of entity names with the copy mechanism in our future work.

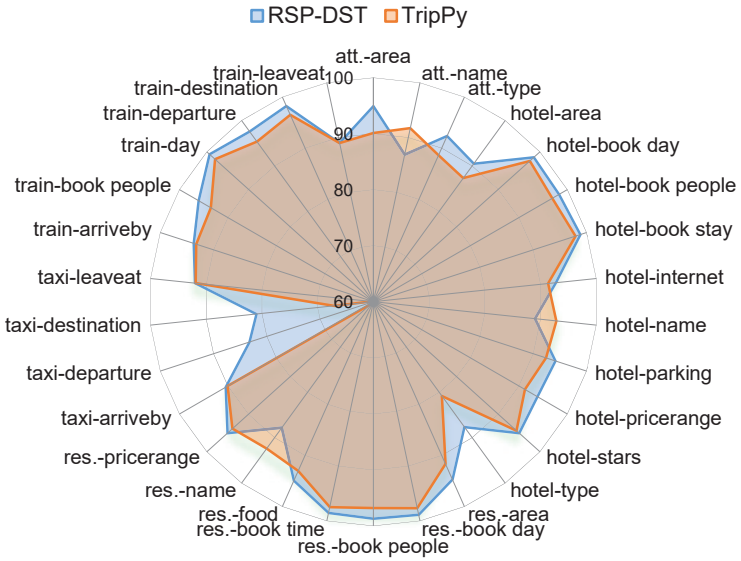


Figure 6. Per-slot accuracy on the test set of MultiWOZ 2.1. In Figure, the "*attraction*" domain is represented as "*att.*" for short and the "*restaurant*" domain is represented as "*res.*".

5.3. Each Turn Joint Goal Accuracy

Usually, the conversation with a longer turn is likely to be more difficult to correctly predict the dialogue state of every turn. The reason is that the longer the turn, the more complex the conversation, and the more dialogue history needs to be taken into account when predicting the dialogue state. Furthermore, the errors of the dialogue state at the current turn are more likely to be carried over to the next turn. Figure 7 shows the performance of RSP-DST at different conversation depths. The histogram in Figure 7 represents the proportion of conversations of different depths in MultiWOZ 2.1 with more than 60% of the dialogues having a conversation depth of six or more turns. It is obvious that the joint goal accuracy decreases as the number of dialogue turns increase. Additionally, we observe that the accuracy of RSP-DST achieves great improvement based on the RSP-DST-base at each turn due to the revising prediction process, effectively alleviating error propagation. The excellent performance on different dialogue turns further illustrates the effectiveness of our approach.

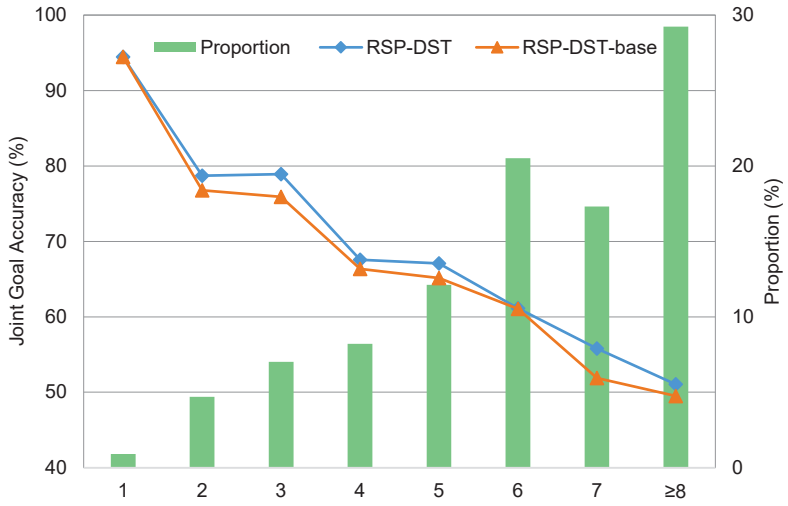


Figure 7. Joint goal accuracy at each turn on the test set of MultiWOZ 2.1.

5.4. Effect of Dialogue History and Previous Dialogue State

We design four types of input sequences: only current turn dialogue (D_t), previous dialogue state and current turn (D_t, B_{t-1}), history and current turn dialogue (D_t, P_{t-1}), previous dialogue state and context (composed of dialogue history and current turn) (D_t, P_{t-1}, B_{t-1}). The results are provided in Table 3. It can be found that the structures with previous dialogue states outperform the other structures. Furthermore, simultaneously taking the previous dialogue state, history, and current turn as part of the input sequence can lead to a better result. In addition, if only the current turn dialogue is used, the performance of RSP-DST drops significantly, with only 18.96% joint goal accuracy. Figure 8 shows the training loss of different structures in the training set of MultiWOZ 2.1. By simultaneously using the previous dialogue state, history and current turn, the best loss can be obtained in a short time, confirming that we have made the best input sequence composed of the previous dialogue state, history and current turn.

Table 3. Different structures' joint goal accuracy on the test set of MultiWOZ 2.1.

Structure	Joint Goal Accuracy(%)	Slot Accuracy(%)
$(B_t \mid D_t)$	18.96	86.33
$(B_t \mid D_t, B_{t-1})$	56.43	97.63
$(B_t \mid D_t, P_{t-1})$	55.85	96.68
$(B_t \mid D_t, P_{t-1}, B_{t-1})$	**58.09**	**97.75**

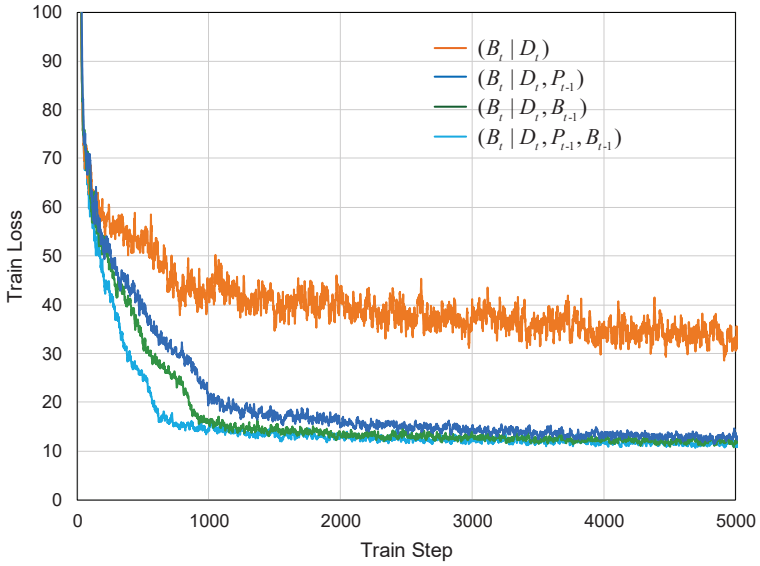


Figure 8. Train loss of different structures on the training set of MultiWOZ 2.1.

5.5. Error Analysis

Finally, we further explore the error rate of each slot. Figure 9 shows the results of RSP-DST-base and RSP-DST on the test set of MultiWOZ 2.1. From Figure 9, we notice that the error rates of most slots significantly reduce due to the revising prediction process, which further demonstrates the effectiveness of our approach. We also observe that error rates of some number-related slots such as "*hotel-book stay*" in RSP-DST are higher than RSP-DST-base. This is probably because the values of these slots are confusing. In addition, we find that error rates of some place-related slots such as "*attraction-name*" are at a high level, though error rates have reduced with the help of the revising prediction process. Values of these slots are multifarious and some of them never appear in the training set. In terms of unseen values, adopting an open vocabulary-based method would be efficient. This observation encourages us to enhance the revising prediction process with open vocabulary in our future work.

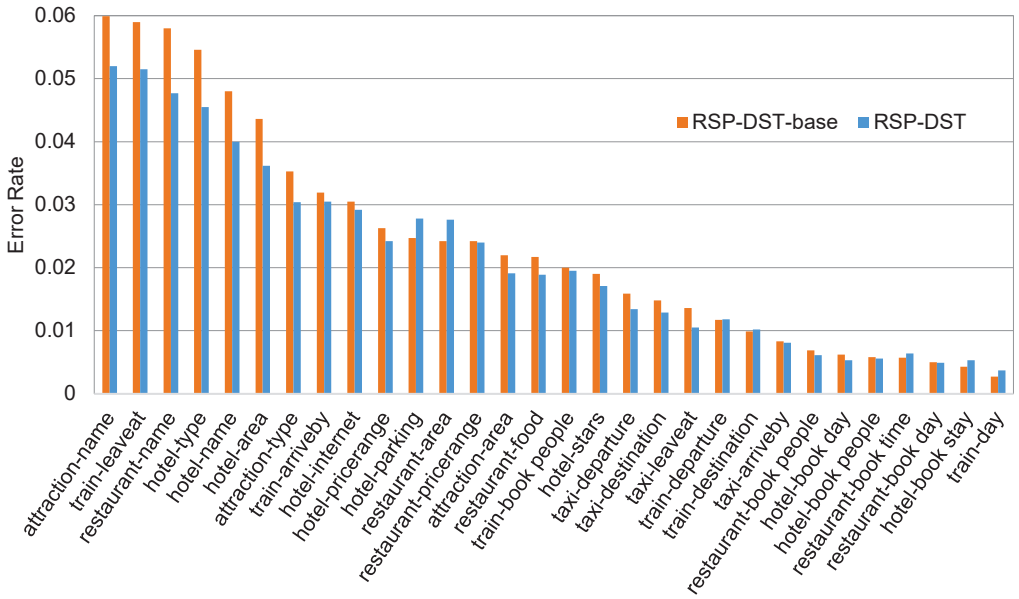
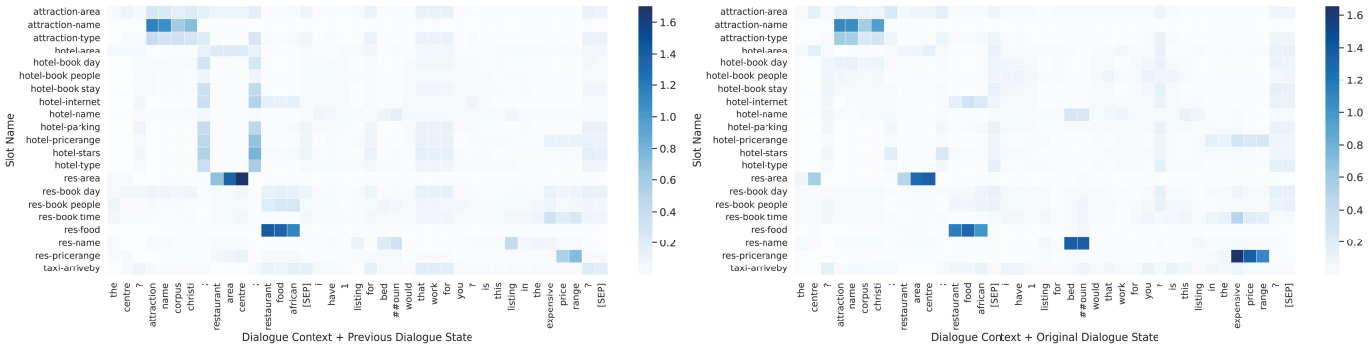


Figure 9. The error rate of every slot on the test set of MultiWOZ 2.1.

5.6. Visualization

The results in Table 2 show that the revising prediction achieves measurable improvements (1.93, 1.78 and 1.01%, respectively, in joint goal accuracy) over the original prediction. To further demonstrate the effectiveness of the revising prediction, Figure 10 visualizes the prediction process at turn four on an example dialogue ID PMUL2279 (Figure 1) from MultiWOZ 2.1. In this example, dialogue involves two domains (*attraction* and *restaurant*), slots *restaurant-name* and *restaurant-pricerange* are new at turn four by "**System**: I have 1 listing for bedouin would that work for you? **User**: Is this listing in the expensive price range?". The ground truth labels of *restaurant-name* and *restaurant-pricerange* are *bedouin* and *expensive*, respectively.



(**a**) Original Prediction (**b**) Revising Prediction

Figure 10. Visualization of the (**a**) original prediction and (**b**) revising prediction at turn four on an example of dialogue ID PMUL2279 (Figure 1) from MultiWOZ 2.1. Slots *restaurant-name* and *restaurant-pricerange* are new at turn turn, and the ground truth labels of *restaurant-name* and *restaurant-pricerange* are *bedouin* and *expensive*, respectively. In both figures, the ordinate is the slot name, the abscissa is the input sequence, and "*res*" represents "*restaurant*". Due to space limitations, the ordinate and abscissa represent only a fraction of the entire data.

In the original prediction process, the slot values of *attraction-name*, *restaurant-area*, and *restaurant-food* are obtained from the previous dialogue state. However, for *restaurant-name* and *restaurant-pricerange*, the model pays little attention to their values in the dialogue context at turn four, indicating that the model thinks that the dialogue context has nothing

to do with the slots *restaurant-name* and *restaurant-pricerange* and then predicts their values as *none*. In addition, we find that the model pays more attention to the previous dialogue state for hotel-related slots which are never mentioned in the dialogue context. In the revising prediction process, RSP-DST jointly models the original dialogue state and context in order to detect and revise errors existing in the original prediction process. Consequently, slots *restaurant-name* and *restaurant-pricerange* assign high attention weights to their values in the dialogue context and hotel-related slots which are irrelevant to this dialogue and successfully reduce the focus on the dialogue state. This result demonstrates that the revising prediction process can make use of the original dialogue state and context, which can be considered as a reference, to revise errors in the original dialogue state.

6. Conclusions

In this paper, we propose a novel DST model RSP-DST, which constructs a two-stage slot value prediction process composed of an original and a revising prediction. In the original prediction process, RSP-DST jointly models the previous dialogue state and context to predict the original dialogue state of the current conversation turn. Then, the revising prediction process utilizes the dialogue context to revise errors existing in the original dialogue state to avoid the errors carried over to the next turn, alleviating error propagation. Comprehensive experiments were conducted on MultiWOZ 2.0, MultiWOZ 2.1, and MultiWOZ 2.4 and the results indicate that RSP-DST outperforms previous SOTA works and achieves a new SOTA performance. In our future work, we intend to enhance the extraction of entity names with the copy mechanism based on open vocabulary and explore a novel generative method to improve the performance of DST with knowledge graphs in the open domain.

Author Contributions: Conceptualization, Q.L. and W.Z.; methodology, Q.L.; software, Q.L.; validation, Q.L.; formal analysis, Q.L. and W.Z.; investigation, Q.L.; resources, W.Z. and M.H.; data curation, S.F. and Y.W.; writing—original draft preparation, Q.L.; writing—review and editing, Q.L., W.Z. and M.H.; visualization, Q.L.; supervision, W.Z. and S.F.; project administration, W.Z., M.H., S.F. and Y.W.; funding acquisition, W.Z. and M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by The National Natural Science Foundation of China (Nos. 82260362, 62241202) and and The National Key R&D Program of China (No. 2021ZD0111000).

Data Availability Statement: The MultiWOZ 2.0 (accessed on 23 April 2022) dataset analysed during the current study are available in https://www.repository.cam.ac.uk/bitstream/handle/1810/280608/MULTIWOZ2.zip?sequence=3&isAllowed=y, MultiWOZ 2.1 (accessed on 23 April 2022) dataset are available in https://www.repository.cam.ac.uk/bitstream/handle/1810/294507/MULTIWOZ2.1.zip?sequence=1&isAllowed=y, and MultiWOZ 2.4 (accessed on 12 August 2022) dataset are available in https://github.com/smartyfh/MultiWOZ2.4/blob/main/data/MULTIWOZ2.4.zip (accessed on 1 January 2020).

Acknowledgments: The authors appreciate the reviewers and editors for their useful comments and work on this article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Description of Symbols

Table A1. Description of symbols in this paper.

Symbol	Description
$\mathcal{D}$	A set of dialogues with T turns
D_t	A dialogue at turn t consisting of system response and user utterance
R_t	System response at turn t
U_t	User utterance at turn t
P_{t-1}	The dialogue history of turn t

Table A1. *Cont.*

Symbol	Description
$\mathcal{S}$	A set of M predefined slots
S_m	The m-th slot in $\mathcal{S}$
V_m^t	The corresponding value of slot S_m at turn t
B_t	The dialogue state at turn t consisting of a set of (*slot*, *value*) pairs
X_t	The input sequence of dialogue context at turn t
$\boldsymbol{H}_t$	The output of $BERT_{finetune}$, and it is the matrix form of all tokens' representations in X_t
$\boldsymbol{h}_{s_m}^{[CLS]}$	The output of $BERT_{fixed}$, and it is the is vector representation of slot S_m
$\boldsymbol{h}_{s_m,t}$	The slot attention vector of slot S_m at turn t
$\tilde{\boldsymbol{h}}_{s_m,t}$	The token-aware slot vector representation of slot S_m at turn t
$\tilde{\boldsymbol{H}}_{s,t}$	The matrix form of all slots' vector representations at turn t
$\hat{\boldsymbol{h}}_{s_m,t}$	The slot-related representation of slot S_m at turn t
$\hat{\boldsymbol{H}}_{s,t}$	The matrix form of all slot-related representations at turn t
$\boldsymbol{o}_{s_m}^t$	The final semantic vector representation of slot S_m at turn t

References

1. Chen, H.; Liu, X.; Yin, D.; Tang, J. A Survey on Dialogue Systems: Recent Advances and New Frontiers. *SIGKDD Explor. Newsl.* **2017**, *19*, 25–35. [CrossRef]
2. Gao, J.; Galley, M.; Li, L. Neural Approaches to Conversational AI. *Found. Trends® Inf. Retr.* **2019**, *13*, 127–298. [CrossRef]
3. Ni, J.; Young, T.; Pandelea, V.; Xue, F.; Cambria, E. Recent advances in deep learning based dialogue systems: A systematic survey. *Artif. Intell. Rev.* **2023**, *56*, 3055–3155. [CrossRef]
4. Mrkšić, N.; Ó Séaghdha, D.; Wen, T.H.; Thomson, B.; Young, S. Neural Belief Tracker: Data-Driven Dialogue State Tracking. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 1777–1788.
5. Williams, J.D.; Raux, A.; Henderson, M. The dialog state tracking challenge series: A review. *Dialogue Discourse* **2016**, *7*, 4–33. [CrossRef]
6. Chen, Y.N.; Celikyilmaz, A.; Hakkani-Tür, D. Deep Learning for Dialogue Systems. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 8–14.
7. Jacqmin, L.; Rojas Barahona, L.M.; Favre, B. "Do you follow me?": A Survey of Recent Approaches in Dialogue State Tracking. In Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue, Edinburgh, UK, 7–9 September 2022; pp. 336–350.
8. Balaraman, V.; Sheikhalishahi, S.; Magnini, B. Recent Neural Methods on Dialogue State Tracking for Task-Oriented Dialogue Systems: A Survey. In Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, Singapore, 29–31 July 2021; pp. 239–251.
9. Zhong, V.; Xiong, C.; Socher, R. Global-Locally Self-Attentive Encoder for Dialogue State Tracking. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 1458–1467.
10. Lee, H.; Lee, J.; Kim, T.Y. SUMBT: Slot-Utterance Matching for Universal and Scalable Belief Tracking. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5478–5483.
11. Zhu, S.; Li, J.; Chen, L.; Yu, K. Efficient Context and Schema Fusion Networks for Multi-Domain Dialogue State Tracking. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 766–781.
12. Ye, F.; Manotumruksa, J.; Zhang, Q.; Li, S.; Yilmaz, E. Slot Self-Attentive Dialogue State Tracking. In Proceedings of the Web Conference 2021, WWW'21, Ljubljana, Slovenia, 19–23 April 2021; pp. 1598–1608.
13. Heck, M.; van Niekerk, C.; Lubis, N.; Geishauser, C.; Lin, H.C.; Moresi, M.; Gašić, M. TripPy: A Triple Copy Strategy for Value Independent Neural Dialog State Tracking. In Proceedings of the 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue, Virtual meeting, 1–3 July 2020; pp. 35–44.
14. Wu, C.S.; Madotto, A.; Hosseini-Asl, E.; Xiong, C.; Socher, R.; Fung, P. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 808–819.
15. Kim, S.; Yang, S.; Kim, G.; Lee, S.W. Efficient Dialogue State Tracking by Selectively Overwriting Memory. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 567–582.
16. Zeng, Y.; Nie, J. Multi-Domain Dialogue State Tracking—A Purely Transformer-Based Generative Approach. *arXiv* **2020**, arXiv:2010.14061.
17. Budzianowski, P.; Wen, T.H.; Tseng, B.H.; Casanueva, I.; Ultes, S.; Ramadan, O.; Gašić, M. 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, Brussels, Belgium, 31 October–4 November 2018; pp. 5016–5026.

18. Eric, M.; Goel, R.; Paul, S.; Sethi, A.; Agarwal, S.; Gao, S.; Kumar, A.; Goyal, A.; Ku, P.; Hakkani-Tur, D. 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, Marseille, France, 11–16 May 2020; pp. 422–428.
19. Ye, F.; Manotumruksa, J.; Yilmaz, E. MultiWOZ 2.4: A Multi-Domain Task-Oriented Dialogue Dataset with Essential Annotation Corrections to Improve State Tracking Evaluation. In Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue, Edinburgh, UK, 7–9 September 2022; pp. 351–360.
20. Thomson, B.; Young, S. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Comput. Speech Lang.* **2010**, *24*, 562–588. [CrossRef]
21. Henderson, M.; Thomson, B.; Young, S. Word-Based Dialog State Tracking with Recurrent Neural Networks. In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), Philadelphia, PA, USA, 18–20 June 2014; pp. 292–299.
22. Williams, J.D. Web-style ranking and SLU combination for dialog state tracking. In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), Philadelphia, PA, USA, 18–20 June 2014; pp. 282–291.
23. Wen, T.H.; Vandyke, D.; Mrkšić, N.; Gašić, M.; Rojas-Barahona, L.M.; Su, P.H.; Ultes, S.; Young, S. A Network-based End-to-End Trainable Task-oriented Dialogue System. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017; Volume 1, pp. 438–449.
24. He, Y.; Tang, Y. A Neural Language Understanding for Dialogue State Tracking. In *Proceedings of the Knowledge Science, Engineering and Management*; Springer International Publishing: Cham, Switzerland, 2021; pp. 542–552.
25. Rastogi, P.; Gupta, A.; Chen, T.; Lambert, M. Scaling Multi-Domain Dialogue State Tracking via Query Reformulation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 2, pp. 97–105.
26. Ren, L.; Ni, J.; McAuley, J. Scalable and Accurate Dialogue State Tracking via Hierarchical Sequence Generation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 1876–1885.
27. Ren, L.; Xie, K.; Chen, L.; Yu, K. Towards Universal Dialogue State Tracking. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2780–2786.
28. Ouyang, Y.; Chen, M.; Dai, X.; Zhao, Y.; Huang, S.; Chen, J. Dialogue State Tracking with Explicit Slot Connection Modeling. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 34–40.
29. Feng, Y.; Wang, Y.; Li, H. 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, Online, 1–6 August 2021; Volume 1, pp. 1714–1725.
30. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
31. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. *OpenAI blog* **2019**, *1*, 9.
32. Shan, Y.; Li, Z.; Zhang, J.; Meng, F.; Feng, Y.; Niu, C.; Zhou, J. A contextual hierarchical attention network with adaptive objective for dialogue state tracking. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6322–6333.
33. Zhang, J.; Hashimoto, K.; Wu, C.S.; Wang, Y.; Yu, P.; Socher, R.; Xiong, C. 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, Barcelona, Spain, 13–14 September 2020; pp. 154–167.
34. Hosseini-Asl, E.; McCann, B.; Wu, C.S.; Yavuz, S.; Socher, R. A Simple Language Model for Task-Oriented Dialogue. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 20179–20191.
35. Pan, B.; Yang, Y.; Li, B.; Cai, D. Self-supervised attention flow for dialogue state tracking. *Neurocomputing* **2021**, *440*, 279–286. [CrossRef]
36. Wang, Y.; He, T.; Mei, J.; Fan, R.; Tu, X. A Stack-Propagation Framework with Slot Filling for Multi-Domain Dialogue State Tracking. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–15. [CrossRef] [PubMed]
37. Zhu, Q.; Li, B.; Mi, F.; Zhu, X.; Huang, M. Continual Prompt Tuning for Dialog State Tracking. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Volume 1, pp. 1124–1137.
38. Chen, D. Neural Reading Comprehension and Beyond. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2018.
39. Wilcock, G.; Jokinen, K. Conversational AI and Knowledge Graphs for Social Robot Interaction. In Proceedings of the 2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Sapporo, Japan, 7–10 March 2022; pp. 1090–1094.
40. Shen, Y.; Ding, N.; Zheng, H.T.; Li, Y.; Yang, M. Modeling Relation Paths for Knowledge Graph Completion. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 3607–3617. [CrossRef]
41. Wu, Y.; Liao, L.; Zhang, G.; Lei, W.; Zhao, G.; Qian, X.; Chua, T.S. State Graph Reasoning for Multimodal Conversational Recommendation. *IEEE Trans. Multimed.* **2022**. [CrossRef]

42. Zhou, L.; Small, K. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *arXiv* **2019**, arXiv:1911.06192.
43. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
44. Chen, L.; Lv, B.; Wang, C.; Zhu, S.; Tan, B.; Yu, K. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7521–7528.
45. Zhao, M.; Wang, L.; Jiang, Z.; Li, R.; Lu, X.; Hu, Z. Multi-task learning with graph attention networks for multi-domain task-oriented dialogue systems. *Knowl.-Based Syst.* **2023**, *259*, 110069. [CrossRef]
46. Feng, Y.; Lipani, A.; Ye, F.; Zhang, Q.; Yilmaz, E. Dynamic Schema Graph Fusion Network for Multi-Domain Dialogue State Tracking. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Volume 1, pp. 115–126.
47. Moerland, T.M.; Broekens, J.; Plaat, A.; Jonker, C.M. Model-based Reinforcement Learning: A Survey. *Found. Trends® Mach. Learn.* **2023**, *16*, 1–118. [CrossRef]
48. Huang, Y.; Feng, J.; Hu, M.; Wu, X.; Du, X.; Ma, S. Meta-Reinforced Multi-Domain State Generator for Dialogue Systems. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7109–7118.
49. Gao, S.; Sethi, A.; Agarwal, S.; Chung, T.; Hakkani-Tur, D. Dialog State Tracking: A Neural Reading Comprehension Approach. In Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, Stockholm, Sweden, 11–13 September 2019; pp. 264–273.
50. Gao, S.; Agarwal, S.; Jin, D.; Chung, T.; Hakkani-Tur, D. From Machine Reading Comprehension to Dialogue State Tracking: Bridging the Gap. In Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, Online, 5–10 July 2020; pp. 79–89.
51. Hu, J.; Yang, Y.; Chen, C.; He, L.; Yu, Z. SAS: Dialogue State Tracking via Slot Attention and Slot Information Sharing. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6366–6375.
52. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. In Proceedings of the *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 5998–6008.
53. Lei Ba, J.; Kiros, J.R.; Hinton, G.E. Layer Normalization. *arXiv* **2016**, arXiv:1607.06450.
54. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2017**, arXiv:1711.05101.
55. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
56. Bowman, S.; Vilnis, L.; Vinyals, O.; Dai, A.; Jozefowicz, R.; Bengio, S. Generating sentences from a continuous space. In Proceedings of the CoNLL 2016—20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; pp. 10–21.