Efficient Dialogue State Tracking by Selectively Overwriting Memory

Sungdong Kim **Sohee Yang** **Gyuwan Kim** **Sang-Woo Lee**
Clova AI, NAVER Corp.
{sungdong.kim, sh.yang, gyuwan.kim, sang.woo.lee}@navercorp.com

Abstract

Recent works in dialogue state tracking (DST) focus on an open vocabulary-based setting to resolve scalability and generalization issues of the predefined ontology-based approaches. However, they are inefficient in that they predict the dialogue state at every turn from scratch. Here, we consider dialogue state as an explicit fixed-sized memory and propose a selectively overwriting mechanism for more efficient DST. This mechanism consists of two steps: (1) predicting state operation on each of the memory slots, and (2) overwriting the memory with new values, of which only a few are generated according to the predicted state operations. Our method decomposes DST into two sub-tasks and guides the decoder to focus only on one of the tasks, thus reducing the burden of the decoder. This enhances the effectiveness of training and DST performance. Our SOM-DST (Selectively Overwriting Memory for Dialogue State Tracking) model achieves state-of-the-art joint goal accuracy with 51.72% in MultiWOZ 2.0 and 53.01% in MultiWOZ 2.1 in an open vocabulary-based DST setting. In addition, we analyze the accuracy gaps between the current and the ground truth-given situations and suggest that it is a promising direction to improve state operation prediction to boost the DST performance.[1]

1 Introduction

Building robust task-oriented dialogue systems has gained increasing popularity in both the research and industry communities (Chen et al., 2017). Dialogue state tracking (DST), one of the essential tasks in task-oriented dialogue systems (Zhong et al., 2018), is keeping track of user goals or intentions throughout a dialogue in the form of a set of slot-value pairs, i.e., dialogue state. Because the next dialogue system action is selected based on the current dialogue state, an accurate prediction of the dialogue state has significant importance.

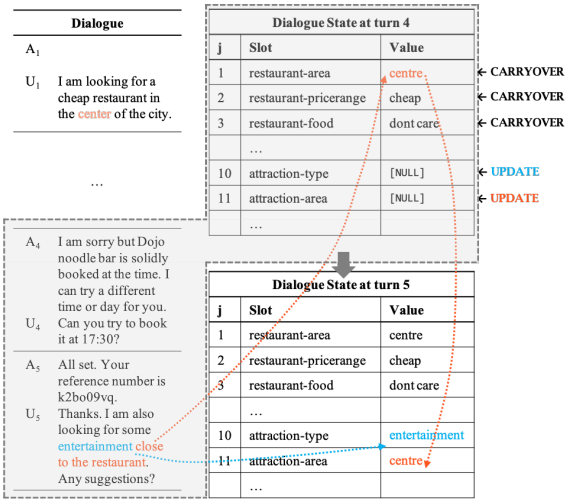

Figure 1: An example of how SOM-DST performs dialogue state tracking at a specific dialogue turn (in this case, fifth). The shaded part is the input to the model, and "Dialogue State at turn 5" at the right-bottom part is the output of the model. Here, **UPDATE** operation needs to be performed on the 10th and 11th slot. DST at this turn is challenging since the model requires reasoning over the long-past conversation. However, SOM-DST can still robustly perform DST because the previous dialogue state is directly utilized like a memory.

Traditional neural DST approaches assume that all candidate slot-value pairs are given in advance, i.e., they perform predefined ontology-based DST (Mrkšić et al., 2017; Zhong et al., 2018; Nouri and Hosseini-Asl, 2018; Lee et al., 2019). Most previous works that take this approach perform DST by scoring all possible slot-value pairs in the ontology and selecting the value with the highest score as the predicted value of a slot. Such an approach has been widely applied to datasets like DSTC2 and WOZ2.0, which have a relatively small ontology size. (Henderson et al., 2014; Wen et al., 2017)

[1]The code is available at github.com/clovaai/som-dst.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 567–582

Although this approach simplifies the task, it has inherent limitations: (1) it is often difficult to obtain the ontology in advance, especially in a real scenario (Xu and Hu, 2018), (2) predefined ontology-based DST cannot handle previously unseen slot values, and (3) the approach does not scale large since it has to go over all slot-value candidates at every turn to predict the current dialogue state. Indeed, recent DST datasets often have a large size of ontology; e.g., the total number of slot-value candidates in MultiWOZ 2.1 is 4510, while the numbers are much smaller in DSTC2 and WOZ2.0 as 212 and 99, respectively (Budzianowski et al., 2018).

To address these issues, recent methods employ an approach that either directly generates or extracts a value from the dialogue context for every slot, allowing open vocabulary-based DST (Lei et al., 2018; Gao et al., 2019; Wu et al., 2019; Ren et al., 2019). While this formulation is relatively more scalable and robust to handling unseen slot values, many of the previous works do not efficiently perform DST since they predict the dialogue state from scratch at every dialogue turn.

In this work, we focus on an open vocabulary-based setting and propose SOM-DST (Selectively Overwriting Memory for Dialogue State Tracking). Regarding dialogue state as a memory that can be selectively overwritten (Figure 1), SOM-DST decomposes DST into two sub-tasks: (1) state operation prediction, which decides the types of the operations to be performed on each of the memory slots, and (2) slot value generation, which generates the values to be newly written on a subset of the memory slots (Figure 2). This decomposition allows our model to efficiently generate the values of only a minimal subset of the slots, while many of the previous works generate or extract the values of all slots at every dialogue turn. Moreover, this decomposition reduces the difficulty of DST in an open-vocabulary based setting by clearly separating the roles of the encoder and the decoder. Our encoder, i.e., state operation predictor, can focus on selecting the slots to pass to the decoder so that the decoder, i.e., slot value generator, can focus only on generating the values of those selected slots. To the best of our knowledge, our work is the first to propose such a selectively overwritable memory-like perspective and a discrete two-step approach on DST.

Our proposed SOM-DST achieves state-of-the-art joint goal accuracy in an open vocabulary-based DST setting on two of the most actively studied datasets: MultiWOZ 2.0 and MultiWOZ 2.1. Error analysis (Section 6.2) further reveals that improving state operation prediction can significantly boost the final DST accuracy.

In summary, the contributions of our work built on top of a perspective that considers dialogue state tracking as selectively overwriting memory are as follows:

- Enabling efficient DST, generating the values of a minimal subset of the slots by utilizing the previous dialogue state at each turn.
- Achieving state-of-the-art performance on MultiWOZ 2.0 and MultiWOZ 2.1 in an open vocabulary-based DST setting.
- Highlighting the potential of improving the state operating prediction accuracy in our proposed framework.

2 Previous Open Vocabulary-based DST

Many works on recent task-oriented dialogue datasets with a large scale ontology, such as MultiWOZ 2.0 and MultiWOZ 2.1, solve DST in an open vocabulary-based setting (Gao et al., 2019; Wu et al., 2019; Ren et al., 2019; Le et al., 2020a,b).

Wu et al. (2019) show the potential of applying the encoder-decoder framework (Cho et al., 2014a) to open vocabulary-based DST. However, their method is not computationally efficient because it performs autoregressive generation of the values for all slots at every dialogue turn.

Ren et al. (2019) tackle the drawback of the model of Wu et al. (2019), that their model generates the values of all slots at every dialogue turn, by using a hierarchical decoder. In addition, they come up with a new notion dubbed Inference Time Complexity (ITC) to compare the efficiency of different DST models. ITC is calculated using the number of slots J and the number of corresponding slot values M.[2] Following their work, we also calculate ITC in Appendix B for comparison.

Le et al. (2020b) introduce another work that tackles the efficiency issue. To maximize the computational efficiency, they use a non-autoregressive decoder to generate the slot values of the current dialogue state at once. They encode the slot type information together with the dialogue context and

[2]The notations used in the work of Ren et al. (2019) are n and m, respectively.

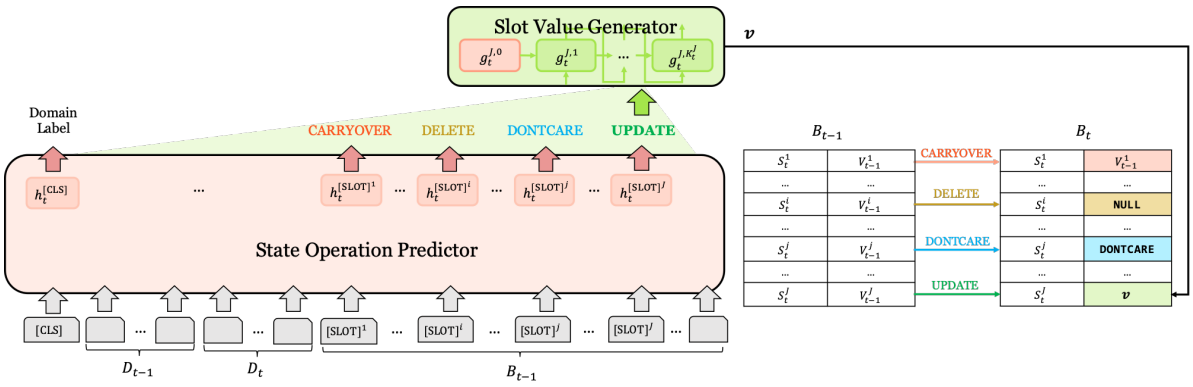

Figure 2: The overview of the proposed SOM-DST. SOM-DST takes the previous turn dialogue utterances D_{t-1}, current turn dialogue utterances D_t, and the previous dialogue state B_{t-1} as the input and outputs the current dialogue state B_t. This is performed by two sub-components: state operation predictor and slot value generator. State operation predictor takes D_{t-1}, D_t, and B_{t-1} as the input and predicts the operations to perform on each of the slots. Domain classification is jointly performed as an auxiliary task. Slot value generator generates the values for the slots that take **UPDATE** as the predicted operation. The value generation for a slot is done in an autoregressive manner.

the delexicalized dialogue context. They do not use the previous turn dialogue state as the input.

Le et al. (2020a) process the dialogue context in both domain-level and slot-level. They make the final representation to generate the values using a late fusion approach. They show that there is a performance gain when the model is jointly trained with response generation. However, they still generate the values of every slot at each turn, like Wu et al. (2019).

Gao et al. (2019) formulate DST as a reading comprehension task and propose a model named DST Reader that extracts the values of the slots from the input. They introduce and show the importance of the concept of a slot carryover module, i.e., a component that makes a binary decision whether to carry the value of a slot from the previous turn dialogue state over to the current turn dialogue state. The definition and use of discrete operations in our work is inspired by their work.

Zhang et al. (2019) target the issue of ill-formatted strings that generative models suffer from. In order to avoid this issue, they take a hybrid approach. For the slots they categorize as picklist-based slots, they use a predefined ontology-based approach as in the work of Lee et al. (2019); for the slots they categorize as span-based slots, they use a span extraction-based method like DST-Reader (Gao et al., 2019). However, their hybrid model shows lower performance than when they use only the picklist-based approach. Although their solely picklist-based model achieves state-of-the-art joint accuracy in MultiWOZ 2.1, it is done in a predefined ontology-based setting, and thus cannot avoid the scalability and generalization issues of predefined ontology-based DST.

3 Selectively Overwriting Memory for Dialogue State Tracking

Figure 2 illustrates the overview of SOM-DST. To describe the proposed SOM-DST, we formally define the problem setting in our work.

Dialogue State We define the dialogue state at turn t, $\mathcal{B}_t = \{(S^j, V_t^j) \mid 1 \leq j \leq J\}$, as a fixed-sized memory whose keys are slots S^j and values are the corresponding slot value V_t^j, where J is the total number of such slots. Following the convention of MultiWOZ 2.0 and MultiWOZ 2.1, we use the term "slot" to refer to the concatenation of a domain name and a slot name.

Special Value There are two special values `NULL` and `DONTCARE`. `NULL` means that no information is given about the slot up to the turn. For instance, the dialogue state before the beginning of any dialogue $\mathcal{B}_0$ has only `NULL` as the value of all slots. `DONTCARE` means that the slot neither needs to be tracked nor considered important in the dialogue at that time.[3]

Operation At every turn t, an operation $r_t^j \in \mathcal{O} = \{\textbf{CARRYOVER}, \textbf{DELETE}, \textbf{DONTCARE}, \textbf{UPDATE}\}$ is chosen by the state operation predictor (Section

[3] Such notions of "none value" and "dontcare value" appear in the previous works as well (Wu et al., 2019; Gao et al., 2019; Le et al., 2020b; Zhang et al., 2019).

3.1) and performed on each slot S^j to set its current turn corresponding value V_t^j. When an operation is performed, it either keeps the slot value unchanged (**CARRYOVER**) or changes it to some value different from the previous one (**DELETE**, **DONTCARE**, and **UPDATE**) as the following.

$$V_t^j = \begin{cases} V_{t-1}^j & \text{if } r_t^j = \textbf{CARRYOVER} \\ \texttt{NULL} & \text{if } r_t^j = \textbf{DELETE} \\ \texttt{DONTCARE} & \text{if } r_t^j = \textbf{DONTCARE} \\ v & \text{if } r_t^j = \textbf{UPDATE} \end{cases}$$

The operations that set the value of a slot to a special value (**DELETE** to NULL and **DONTCARE** to DONTCARE, respectively) are chosen only when the previous slot value V_{t-1}^j is not the corresponding special value. **UPDATE** operation requires the generation of a new value $v \notin \{V_{t-1}^j, \texttt{NULL}, \texttt{DONTCARE}\}$ by slot value generator (Section 3.2).

State operation predictor performs state operation prediction as a classification task, and slot value generator performs slot value generation to find out the values of the slots on which **UPDATE** should be performed. The two components of SOM-DST are jointly trained to predict the current turn dialogue state.

3.1 State Operation Predictor

Input Representation We denote the representation of the dialogue utterances at turn t as $D_t = A_t \oplus \texttt{;} \oplus U_t \oplus \texttt{[SEP]}$, where A_t is the system response and U_t is the user utterance. ; is a special token used to mark the boundary between A_t and U_t, and [SEP] is a special token used to mark the end of a dialogue turn. We denote the representation of the dialogue state at turn t as $B_t = B_t^1 \oplus \ldots \oplus B_t^J$, where $B_t^j = \texttt{[SLOT]}^j \oplus S^j \oplus - \oplus V_t^j$ is the representation of the j-th slot-value pair. – is a special token used to mark the boundary between a slot and a value. $\texttt{[SLOT]}^j$ is a special token used to aggregate the information of the j-th slot-value pair into a single vector, like the use case of [CLS] token in BERT (Devlin et al., 2019). In this work, we use the same special token [SLOT] for all $\texttt{[SLOT]}^j$. Our state operation predictor employs a pretrained BERT encoder. The input tokens to the state operation predictor are the concatenation of the previous turn dialog utterances, the current turn dialog utterances, and the previous turn dialog state:[4]

$$X_t = \texttt{[CLS]} \oplus D_{t-1} \oplus D_t \oplus B_{t-1},$$

where [CLS] is a special token added in front of every turn input. Using the previous dialogue state as the input serves as an explicit, compact, and informative representation of the dialogue history for the model.

When the value of the j-th slot at time $t-1$, i.e., V_{t-1}^j, is NULL, we use a special token [NULL] as the input. When the value is DONTCARE, we use the string "dont care" to take advantage of the semantics of the phrase "don't care" that the pretrained BERT encoder would have already learned.

The input to BERT is the sum of the embeddings of the input tokens X_t, segment id embeddings, and position embeddings. For the segment id, we use 0 for the tokens that belong to D_{t-1} and 1 for the tokens that belong to D_t or B_{t-1}. The position embeddings follow the standard choice of BERT.

Encoder Output The output representation of the encoder is $H_t \in \mathbb{R}^{|X_t| \times d}$, and $h_t^{\texttt{[CLS]}}, h_t^{\texttt{[SLOT]}^j} \in \mathbb{R}^d$ are the outputs that correspond to [CLS] and $\texttt{[SLOT]}^j$, respectively. h_t^X, the aggregated sequence representation of the entire input X_t, is obtained by a feed-forward layer with a learnable parameter $W_{pool} \in \mathbb{R}^{d \times d}$ as:

$$h_t^X = \tanh(W_{pool}\, h_t^{\texttt{[CLS]}}).$$

State Operation Prediction State operation prediction is a four-way classification performed on top of the encoder output for each slot representation $h_t^{\texttt{[SLOT]}^j}$:

$$P_{opr,t}^j = \text{softmax}(W_{opr}\, h_t^{\texttt{[SLOT]}^j}),$$

where $W_{opr} \in \mathbb{R}^{|\mathcal{O}| \times d}$ is a learnable parameter and $P_{opr,t}^j \in \mathbb{R}^{|\mathcal{O}|}$ is the probability distribution over operations for the j-th slot at turn t. In our formulation, $|\mathcal{O}| = 4$, because $\mathcal{O} = \{$**CARRYOVER**, **DELETE**, **DONTCARE**, **UPDATE**$\}$.

Then, the operation is determined by $r_t^j = \text{argmax}(P_{opr,t}^j)$ and the slot value generation is performed on only the slots whose operation is **UPDATE**. We define the set of the slot indices which require the value generation as $\mathbb{U}_t = \{j \mid r_t^j = \textbf{UPDATE}\}$, and its size as $J_t' = |\mathbb{U}_t|$.

[4] We use only the previous turn dialogue utterances D_{t-1} as the dialogue history, i.e., the size of the dialogue history is 1. This is because our model assumes Markov property in dialogues as a part of the input, the previous turn dialogue state B_{t-1}, can serve as a compact representation of the whole dialogue history.

3.2 Slot Value Generator

For each j-th slot such that $j \in \mathbb{U}_t$, the slot value generator generates a value. Our slot value generator differs from the generators of many of the previous works because it generates the values for only J'_t number of slots, not J. In most cases, $J'_t \ll J$, so this setup enables an efficient computation where only a small number of slot values are newly generated.

We use Gated Recurrent Unit (GRU) (Cho et al., 2014b) decoder like Wu et al. (2019). GRU is initialized with $g_t^{j,0} = h_t^{\mathrm{X}}$ and $e_t^{j,0} = h_t^{\texttt{[SLOT]}^j}$, and recurrently updates the hidden state $g_t^{j,k} \in \mathbb{R}^d$ by taking a word embedding $e_t^{j,k}$ as the input until `[EOS]` token is generated:

$$g_t^{j,k} = \mathrm{GRU}(g_t^{j,k-1}, e_t^{j,k}).$$

The decoder hidden state is transformed to the probability distribution over the vocabulary at the k-th decoding step, where $E \in \mathbb{R}^{d_{vcb} \times d}$ is the word embedding matrix shared across the encoder and the decoder, such that d_{vcb} is the vocabulary size.

$$P_{vcb,t}^{j,k} = \mathrm{softmax}(E\ g_t^{j,k}) \in \mathbb{R}^{d_{vcb}}.$$

As the work of Wu et al. (2019), we use the soft-gated copy mechanism (See et al., 2017) to get the final output distribution $P_{val,t}^{j,k}$ over the candidate value tokens:

$$P_{ctx,t}^{j,k} = \mathrm{softmax}(H_t\ g_t^{j,k}) \in \mathbb{R}^{|X_t|},$$
$$P_{val,t}^{j,k} = \alpha P_{vcb,t}^{j,k} + (1-\alpha) P_{ctx,t}^{j,k},$$

such that α is a scalar value computed as:

$$\alpha = \mathrm{sigmoid}(W_1\ [g_t^{j,k}; e_t^{j,k}; c_t^{j,k}]),$$

where $W_1 \in \mathbb{R}^{1 \times (3d)}$ is a learnable parameter and $c_t^{j,k} = P_{ctx,t}^{j,k}\ H_t \in \mathbb{R}^d$ is a context vector.

3.3 Objective Function

During training, we jointly optimize both state operation predictor and slot value generator.

State Operation Predictor In addition to the state operation classification, we use domain classification as an auxiliary task to force the model to learn the correlation of slot operations and domain transitions in between dialogue turns. Domain classification is done with a softmax layer on top of h_t^X:

$$P_{dom,t} = \mathrm{softmax}(W_{dom}\ h_t^X),$$

where $W_{dom} \in \mathbb{R}^{d_{dom} \times d}$ is a learnable parameter and $P_{dom,t} \in \mathbb{R}^{d_{dom}}$ is the probability distribution over domains at turn t. d_{dom} is the number of domains defined in the dataset.

The loss for each of state operation classification and domain classification is the average of the negative log-likelihood, as follows:

$$L_{opr,t} = -\frac{1}{J}\sum_{j=1}^{J}(Y_{opr,t}^j)^\intercal \log P_{opr,t}^j,$$
$$L_{dom,t} = -(Y_{dom,t})^\intercal \log P_{dom,t},$$

where $Y_{dom,t} \in \mathbb{R}^{d_{dom}}$ is the one-hot vector for the ground truth domain and $Y_{opr,t}^j \in \mathbb{R}^{|\mathcal{O}|}$ is the one-hot vector for the ground truth operation for the j-th slot.

Slot Value Generator The objective function to train slot value generator is also the average of the negative log-likelihood:

$$L_{svg,t} = -\frac{1}{|\mathbb{U}_t|}\sum_{j \in \mathbb{U}_t}\frac{1}{K_t^j}\sum_{k=1}^{K_t^j}(Y_{val,t}^{j,k})^\intercal \log P_{val,t}^{j,k},$$

where K_t^j is the number of tokens of the ground truth value that needs to be generated for the j-th slot. $Y_{val,t}^{j,k} \in \mathbb{R}^{d_{vcb}}$ is the one-hot vector for the ground truth token that needs to be generated for the j-th slot at the k-th decoding step.

Therefore, the final joint loss $L_{joint,t}$ to be minimized at dialogue turn t is the sum of the losses mentioned above:

$$L_{joint,t} = L_{opr,t} + L_{dom,t} + L_{svg,t}.$$

4 Experimental Setup

4.1 Datasets

We use MultiWOZ 2.0 (Budzianowski et al., 2018) and MultiWOZ 2.1 (Eric et al., 2019) as the datasets in our experiments. These datasets are two of the largest publicly available multi-domain task-oriented dialogue datasets, including about 10,000 dialogues within seven domains. MultiWOZ 2.1 is a refined version of MultiWOZ 2.0 in which the annotation errors are corrected.[5]

Following Wu et al. (2019), we use only five domains (*restaurant*, *train*, *hotel*, *taxi*, *attraction*)

[5] See Table 8 in Appendix A for more details of MultiWOZ 2.1.

excluding *hospital* and *police*.[6] Therefore, the number of domains d_{dom} is 5 and the number of slots J is 30 in our experiments. We use the script provided by Wu et al. (2019) to preprocess the datasets.[7]

4.2 Training

We employ the pretrained BERT-base-uncased model[8] for state operation predictor and one GRU (Cho et al., 2014b) for slot value generator. The hidden size of the decoder is the same as that of the encoder, d, which is 768. The token embedding matrix of slot value generator is shared with that of state operation predictor. We use BertAdam as our optimizer (Kingma and Ba, 2015). We use greedy decoding for slot value generator.

The encoder of state operation predictor makes use of a pretrained model, whereas the decoder of slot value generator needs to be trained from scratch. Therefore, we use different learning rate schemes for the encoder and the decoder. We set the peak learning rate and warmup proportion to 4e-5 and 0.1 for the encoder and 1e-4 and 0.1 for the decoder, respectively. We use a batch size of 32 and set the dropout (Srivastava et al., 2014) rate to 0.1. We also utilize word dropout (Bowman et al., 2016) by randomly replacing the input tokens with the special `[UNK]` token with the probability of 0.1. The max sequence length for all inputs is fixed to 256.

We train state operation predictor and slot value generator jointly for 30 epochs and choose the model that reports the best performance on the validation set. During training, we use the ground truth state operations and the ground truth previous turn dialogue state instead of the predicted ones. When the dialogue state is fed to the model, we randomly shuffle the slot order with a rate of 0.5. This is to make state operation predictor exploit the semantics of the slot names and not rely on the position of the slot tokens or a specific slot order. During inference or when the slot order is not shuffled, the slots are sorted alphabetically. We use teacher forcing 50% of the time to train the decoder.

All experiments are performed on NAVER Smart Machine Learning (NSML) platform (Sung et al., 2017; Kim et al., 2018). All the reported results of SOM-DST are averages over ten runs.

[6]The excluded domains take up only a small portion of the dataset and do not even appear in the test set.

[7]github.com/jasonwu0731/trade-dst

[8]github.com/huggingface/transformers

4.3 Baseline Models

We compare the performance of SOM-DST with both predefined ontology-based models and open vocabulary-based models.

FJST uses a bidirectional LSTM to encode the dialogue history and uses a feed-forward network to predict the value of each slot (Eric et al., 2019).

HJST is proposed together with FJST; it encodes the dialogue history using an LSTM like FJST but uses a hierarchical network (Eric et al., 2019).

SUMBT exploits BERT-base as the encoder for the dialogue context and slot-value pairs. After encoding them, it scores every candidate slot-value pair in a non-parametric manner using a distance measure (Lee et al., 2019).

HyST employs a hierarchical RNN encoder and takes a hybrid approach that incorporates both a predefined ontology-based setting and an open vocabulary-based setting (Goel et al., 2019).

DST Reader formulates the problem of DST as an extractive QA task; it uses BERT-base to make the contextual word embeddings and extracts the value of the slots from the input as a span (Gao et al., 2019).

TRADE encodes the whole dialogue context with a bidirectional GRU and decodes the value for every slot using a copy-augmented GRU decoder (Wu et al., 2019).

COMER uses BERT-large as a feature extractor and a hierarchical LSTM decoder to generate the current turn dialogue state itself as the target sequence (Ren et al., 2019).

NADST uses a Transformer-based non-autoregressive decoder to generate the current turn dialogue state (Le et al., 2020b).

ML-BST uses a Transformer-based architecture to encode the dialogue context with the domain and slot information and combines the outputs in a late fusion approach. Then, it generates the slot values and the system response jointly (Le et al., 2020a).

DS-DST uses two BERT-base encoders and takes a hybrid approach of predefined ontology-based DST and open vocabulary-based DST. It defines picklist-based slots for classification similarly to SUMBT and span-based slots for span extraction like DST Reader (Zhang et al., 2019).

Table 1: Joint goal accuracy on the test set of MultiWOZ 2.0 and 2.1. * indicates a result borrowed from Eric et al. (2019). HyST and DS-DST use a hybrid approach, partially taking advantage of the predefined ontology. † indicates the case where BERT-large is used for our model.

	MultiWOZ 2.0	MultiWOZ 2.1
Predefined Ontology		
HJST* (Eric et al., 2019)	38.40	35.55
FJST* (Eric et al., 2019)	40.20	38.00
SUMBT (Lee et al., 2019)	42.40	-
HyST* (Goel et al., 2019)	42.33	38.10
DS-DST (Zhang et al., 2019)	-	51.21
DST-picklist (Zhang et al., 2019)	-	**53.30**
Open Vocabulary		
DST Reader* (Gao et al., 2019)	39.41	36.40
TRADE* (Wu et al., 2019)	48.60	45.60
COMER (Ren et al., 2019)	48.79	-
NADST (Le et al., 2020b)	50.52	49.04
ML-BST (Le et al., 2020a)	-	50.91
SOM-DST (ours)	**51.72**	**53.01**
SOM-DST† (ours)	**52.32**	**53.68**

DST-picklist is proposed together with DS-DST and uses a similar architecture, but it performs only predefined ontology-based DST considering all slots as picklist-based slots (Zhang et al., 2019).

5 Experimental Results

5.1 Joint Goal Accuracy

Table 1 shows the joint goal accuracy of SOM-DST and other models on the test set of MultiWOZ 2.0 and MultiWOZ 2.1. Joint goal accuracy is an accuracy which checks whether all slot values predicted at a turn exactly match the ground truth values.

As shown in the table, SOM-DST achieves state-of-the-art performance in an open vocabulary-based setting. Interestingly, on the contrary to the previous works, our model achieves higher performance on MultiWOZ 2.1 than on MultiWOZ 2.0. This is presumably because our model, which explicitly uses the dialogue state labels as input, benefits more from the error correction on the state annotations done in MultiWOZ 2.1.[9]

[9]Eric et al. (2019) report that the correction of the annotations done in MultiWOZ 2.1 changes about 32% of the state annotations of MultiWOZ 2.0, which indicates that MultiWOZ 2.0 consists of many annotation errors.

Table 2: Domain-specific results on the test set of MultiWOZ 2.1. Our model outperforms other models in *taxi* and *train* domains.

Domain	Model	Joint Accuracy	Slot Accuracy
Attraction	NADST	66.83	98.79
	ML-BST	**70.78**	99.06
	SOM-DST (ours)	69.83	98.86
Hotel	NADST	48.76	97.70
	ML-BST	49.52	97.50
	SOM-DST (ours)	**49.53**	97.35
Restaurant	NADST	65.37	98.78
	ML-BST	**66.50**	98.76
	SOM-DST (ours)	65.72	98.56
Taxi	NADST	33.80	96.69
	ML-BST	23.05	96.42
	SOM-DST (ours)	**59.96**	98.01
Train	NADST	62.36	98.36
	ML-BST	65.12	90.22
	SOM-DST (ours)	**70.36**	98.67

5.2 Domain-Specific Accuracy

Table 2 shows the domain-specific results of our model and the concurrent works which report such results (Le et al., 2020a,b). Domain-specific accuracy is the accuracy measured on a subset of the predicted dialogue state, where the subset consists of the slots specific to a domain.

While the performance is similar to or a little lower than that of other models in other domains, SOM-DST outperforms other models in *taxi* and *train* domains. This implies that the state-of-the-art joint goal accuracy of our model on the test set comes mainly from these two domains.

A characteristic of the data from these domains is that they consist of challenging conversations; the slots of these domains are filled with more diverse values than other domains,[10] and there are more than one domain changes, i.e., the user changes the conversation topic during a dialogue more than once. For a specific example, among the dialogues where the domain switches more than once, the number of conversations that end in *taxi* domain is ten times more than in other cases. A more detailed statistics are given in Table 10 in Appendix A.

Therefore, we assume our model performs relatively more robust DST in such challenging conversations. We conjecture that this strength attributes to the effective utilization of the previous turn dialogue state in its explicit form, like using a memory;

[10]The statistics of the slot value vocabulary size are shown in Table 9 in Appendix A.

Table 3: Joint goal accuracy on the MultiWOZ 2.1 test set when the four-way state operation prediction changes to two-way, three-way, or six-way.

	State Operations	Joint Accuracy
4	**CARRYOVER, DELETE, DONTCARE, UPDATE**	53.01
2	**CARRYOVER, NON-CARRYOVER**	52.06
3	**CARRYOVER, DONTCARE, UPDATE**	52.63
3	**CARRYOVER, DELETE, UPDATE**	52.64
6	**CARRYOVER, DELETE, DONTCARE, UPDATE, YES, NO**	52.97

Table 4: Joint goal accuracy of the current and the ground truth-given situations. Relative error rate is the proportion of the error when 100% is set as the error where no ground truth is used for SOP and SVG. (GT: Ground Truth, SOP: State Operation Prediction, SVG: Slot Value Generation, Pred: Predicted)

	GT SOP	GT SVG	Joint Accuracy	Relative Error Rate
Pred B_{t-1} (w/ Error Propagation)			53.01	100.0
		✓	56.37	92.85
	✓		89.85	21.60
	✓	✓	100.0	0.00
GT B_{t-1} (w/o Error Propagation)			81.00	100.0
		✓	82.80	90.53
	✓		96.27	19.63
	✓	✓	100.0	0.00

the model can explicitly keep even the information mentioned near the beginning of the conversation and directly copy the values from this memory whenever necessary. Figure 1 shows an example of a complicated conversation in MultiWOZ 2.1, where our model accurately predicts the dialogue state. More sample outputs of SOM-DST are provided in Appendix C.

6 Analysis

6.1 Choice of State Operations

Table 3 shows the joint goal accuracy where the four-way state operation prediction changes to two-way, three-way, or six-way.

The joint goal accuracy drops when we use two-way state operation prediction, which is a binary classification of whether to (1) carry over the previous slot value to the current turn or (2) generate a new value, like Gao et al. (2019). We assume the reason is that it is better to separately model operations **DELETE**, **DONTCARE**, and **UPDATE** that correspond to the latter class of the binary classification, since the values of **DELETE** and **DONTCARE** tend to appear implicitly while the values for **UPDATE** are often explicitly expressed in the dialogue.

We also investigate the performance when only three operations are used or two more state operations, **YES** and **NO**, are used. **YES** and **NO** represent the cases where yes or no should be filled as the slot value, respectively. The performance drops in all of the cases.

6.2 Error Analysis

Table 4 shows the joint goal accuracy of the combinations of the cases where the ground truth is used or not for each of the previous turn dialogue state, state operations at the current turn, and slot values for **UPDATE** at the current turn. From this result, we analyze which of state operation predictor and slot value generator is more responsible for the error in the joint goal prediction, under the cases where error propagation occurs or not.

Among the absolute error of 46.99% made under the situation that error propagation occurs, i.e., the dialogue state predicted at the previous turn is fed to the model, it could be argued that 92.85% comes from state operation predictor, 21.6% comes from slot value generator, and 14.45% comes from both of the components. This indicates that at least 78.4% to 92.85% of the error comes from state operation predictor, and at least 7.15% to 21.6% of the error comes from slot value generator. [11]

Among the absolute error of 19% made under the error propagation-free situation, i.e., ground truth previous turn dialogue state is fed to the model, it could be argued that 90.53% comes from state operation predictor, 19.63% comes from slot value generator, and 10.16% comes from both of the components. This indicates that at least 80.37% to 90.53% of the error comes from state operation predictor, and at least 9.47% to 19.63% of the error comes from slot value generator.

Error propagation that comes from using the dialogue state predicted at the previous turn increases the error 2.47 ($=\frac{100-53.01}{100-81.00}$) times. Both with and without error propagation, a relatively large amount

[11] The calculation of the numbers in the paragraph is done as follows. (The figures in the paragraph immediately below are calculated in the same way.)

$100 - 53.01 = 46.99$
$(100 - 56.37)/46.99 = 92.85$
$(100 - 89.85)/46.99 = 21.6$
$92.85 + 21.6 - 100 = 14.45$
$92.85 - 14.45 = 78.4$
$21.6 - 14.45 = 7.15$

Table 5: Statistics of the number of state operations and the corresponding F1 scores of our model in MultiWOZ 2.1.

	# Operations			F1 score
Operation Type	Train	Valid	Test	Test
CARRYOVER	1,584,757	212,608	212,297	98.66
UPDATE	61,628	8,287	8,399	80.10
DONTCARE	1,911	155	235	32.51
DELETE	1,224	80	109	2.86

Table 6: The minimum, average, and maximum number of slots whose values are generated at a turn, calculated on the test set of MultiWOZ 2.1.

Model	Min #	Avg #	Max #
TRADE	30	30	30
ML-BST	30	30	30
COMER	0	5.72	18
SOM-DST (ours)	0	1.14	9

Table 7: Average inference time per dialogue turn of MultiWOZ 2.1 test set, measured on Tesla V100 with a batch size of 1. † indicates the case where BERT-large is used for our model.

Model	Joint Accuracy	Latency
TRADE	45.60	340 ms
NADST	49.04	26 ms
SOM-DST (ours)	**53.01**	27 ms
SOM-DST† (ours)	**53.68**	40 ms

of error comes from state operation predictor, implying that a large room for improvement currently exists in this component. Improving the state operation prediction accuracy, e.g., by tackling the class imbalance shown in Table 5, may have the potential to increase the overall DST performance by a large margin.

6.3 Efficiency Analysis

In Table 6, we compare the number of slot values generated at a turn among various open vocabulary-based DST models that use an autoregressive decoder.

The maximum number of slots whose values are generated by our model at a turn, i.e., the number of slots on which **UPDATE** should be performed, is 9 at maximum and only 1.14 on average in the test set of MultiWOZ 2.1.

On the other hand, TRADE and ML-BST generate the values of all the 30 slots at every turn of a dialogue. COMER generates only a subset of the slot values like our model, but it generates the values of all the slots that have a non-`NULL` value at a turn, which is 18 at maximum and 5.72 on average.

Table 7 shows the latency of SOM-DST and several other models. We measure the inference time for a dialogue turn of MultiWOZ 2.1 on Tesla V100 with a batch size of 1. The models used for comparison are those with official public implementations.

It is notable that the inference time of SOM-DST is about 12.5 times faster than TRADE, which consists of only two GRUs. Moreover, the latency of SOM-DST is compatible with that of NADST, which explicitly uses non-autoregressive decoding, while SOM-DST achieves much higher joint goal accuracy. This shows the efficiency of the proposed selectively overwriting mechanism of SOM-DST, which generates only the minimal slot values at a turn.

In Appendix B, we also investigate Inference Time Complexity (ITC) proposed in the work of Ren et al. (2019), which defines the efficiency of a DST model using J, the number of slots, and M, the number of values of a slot.

7 Conclusion

We propose SOM-DST, an open vocabulary-based dialogue state tracker that regards dialogue state as an explicit memory that can be selectively overwritten. SOM-DST decomposes dialogue state tracking into state operation prediction and slot value generation. This setup makes the generation process efficient because the values of only a minimal subset of the slots are generated at each dialogue turn. SOM-DST achieves state-of-the-art joint goal accuracy on both MultiWOZ 2.0 and MultiWOZ 2.1 datasets in an open vocabulary-based setting. SOM-DST effectively makes use of the explicit dialogue state and discrete operations to perform relatively robust DST even in complicated conversations. Further analysis shows that improving state operation prediction has the potential to increase the overall DST performance dramatically. From this result, we propose that tackling DST with our proposed problem definition is a promising future research direction.

Acknowledgments

The authors would like to thank the members of Clova AI for proofreading this manuscript.

References

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo 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 *EMNLP*.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter*, 19(2):25–35.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014a. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014b. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *SIGDIAL*.

Rahul Goel, Shachi Paul, and Dilek Hakkani-Tür. 2019. Hyst: A hybrid approach for flexible and accurate dialogue state tracking. In *Interspeech*.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *SIGDIAL*.

Hanjoo Kim, Minkyu Kim, Dongjoo Seo, Jinwoong Kim, Heungseok Park, Soeun Park, Hyunwoo Jo, KyungHyun Kim, Youngil Yang, Youngkwan Kim, et al. 2018. Nsml: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Hung Le, Doyen Sahoo, Chenghao Liu, Nancy F. Chen, and Steven C.H. Hoi. 2020a. End-to-end multi-domain task-oriented dialogue systems with multi-level neural belief tracker. In *Submitted to ICLR 2020*.

Hung Le, Richard Socher, and Steven C.H. Hoi. 2020b. Non-autoregressive dialog state tracking. In *ICLR*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. Sumbt: Slot-utterance matching for universal and scalable belief tracking. In *ACL*.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Melbourne, Australia. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *ACL*.

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking model. In *2nd Conversational AI workshop on NeurIPS 2018*.

Liliang Ren, Jianmo Ni, and Julian McAuley. 2019. Scalable and accurate dialogue state tracking via hierarchical sequence generation. In *EMNLP-IJCNLP*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.

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

Nako Sung, Minkyu Kim, Hyunwoo Jo, Youngil Yang, Jingwoong Kim, Leonard Lausen, Youngkwan Kim, Gayoung Lee, Donghyun Kwak, Jung-Woo Ha, et al. 2017. Nsml: A machine learning platform that enables you to focus on your models. *arXiv preprint arXiv:1712.05902*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*.

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 *ACL*.

Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *ACL*.

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Philip S. Yu, Richard Socher, and Caiming Xiong. 2019. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *ACL*.

A Data Statistics

Table 8: Data Statistics of MultiWOZ 2.1.

		# of Dialogues			# of Turns		
Domain	Slots	Train	Valid	Test	Train	Valid	Test
Attraction	area, name, type	2,717	401	395	8,073	1,220	1,256
Hotel	price range, type, parking, book stay, book day, book people, area, stars, internet, name	3,381	416	394	14,793	1,781	1,756
Restaurant	food, price range, area, name, book time, book day, book people	3,813	438	437	15,367	1,708	1,726
Taxi	leave at, destination, departure, arrive by	1,654	207	195	4,618	690	654
Train	destination, day, departure, arrive by, book people, leave at	3,103	484	494	12,133	1,972	1,976

Table 9: Statistics of the slot value vocabulary size in MultiWOZ 2.1.

	Slot Value Vocabulary Size		
Slot Name	Train	Valid	Test
taxi-destination	373	213	213
taxi-departure	357	214	203
restaurant-name	202	162	162
attraction-name	186	145	149
train-leaveat	146	69	117
train-arriveby	112	64	101
restaurant-food	111	81	70
taxi-leaveat	105	68	65
hotel-name	93	65	58
restaurant-book time	64	50	51
taxi-arriveby	95	49	46
train-destination	27	25	24
train-departure	34	23	23
attraction-type	31	17	17
train-book people	11	9	9
hotel-book people	8	8	8
restaurant-book people	9	8	8
hotel-book day	13	7	7
hotel-stars	9	7	7
restaurant-book day	10	7	7
train-day	8	7	7
attraction-area	7	6	6
hotel-area	7	6	6
restaurant-area	7	6	6
hotel-book stay	10	5	5
hotel-parking	4	4	4
hotel-pricerange	7	5	4
hotel-type	5	5	4
restaurant-pricerange	5	4	4
hotel-internet	3	3	3

Table 10: Statistics of domain transition in the test set of MultiWOZ 2.1. There are 140 dialogues with more than one domain transition that end with *taxi* domain. The cases where domain switches more than once and ends in *taxi* are shown in bold. The total number of dialogues with more than one domain transition is 175. We can view these as complicated dialogues.

Domain Transition				
First	Second	Third	Fourth	Count
restaurant	train	-	-	87
attraction	train	-	-	80
hotel	-	-	-	71
train	attraction	-	-	71
train	hotel	-	-	70
restaurant	-	-	-	64
train	restaurant	-	-	62
hotel	train	-	-	57
taxi	-	-	-	51
attraction	restaurant	-	-	38
restaurant	attraction	taxi	-	**35**
restaurant	attraction	-	-	31
train	-	-	-	31
hotel	attraction	-	-	27
restaurant	hotel	-	-	27
restaurant	hotel	taxi	-	**26**
attraction	hotel	taxi	-	**24**
attraction	restaurant	taxi	-	**23**
hotel	restaurant	-	-	22
attraction	hotel	-	-	20
hotel	attraction	taxi	-	**16**
hotel	restaurant	taxi	-	**13**
attraction	-	-	-	12
attraction	restaurant	train	-	3
restaurant	hotel	train	-	3
hotel	train	restaurant	-	3
restaurant	train	hotel	-	3
restaurant	taxi	hotel	-	3
attraction	train	restaurant	-	2
train	attraction	restaurant	-	2
attraction	restaurant	hotel	-	2
hotel	train	attraction	-	2
attraction	taxi	hotel	-	1
hotel	taxi	-	-	1
train	hotel	restaurant	-	1
restaurant	taxi	-	-	1
restaurant	train	taxi	-	**1**
hotel	restaurant	train	-	1
hotel	taxi	train	-	1
taxi	attraction	-	-	1
restaurant	train	attraction	-	1
attraction	train	hotel	-	1
attraction	train	taxi	-	**1**
restaurant	attraction	train	-	1
hotel	taxi	attraction	-	1
train	hotel	attraction	-	1
restaurant	taxi	attraction	-	1
hotel	attraction	restaurant	taxi	**1**
attraction	hotel	train	-	1
taxi	restaurant	train	-	1

B Inference Time Complexity (ITC)

Table 11: Inference Time Complexity (ITC) of each model. We report the ITC in both the best case and the worst case for more precise comparison. J indicates the number of slots, and M indicates the number of values of a slot.

	Inference Time Complexity	
Model	Best	Worst
SUMBT	$\Omega(JM)$	$O(JM)$
DS-DST	$\Omega(J)$	$O(JM)$
DST-picklist	$\Omega(JM)$	$O(JM)$
DST Reader	$\Omega(1)$	$O(J)$
TRADE	$\Omega(J)$	$O(J)$
COMER	$\Omega(1)$	$O(J)$
NADST	$\Omega(1)$	$O(1)$
ML-BST	$\Omega(J)$	$O(J)$
SOM-DST(ours)	$\Omega(1)$	$O(J)$

Inference Time Complexity (ITC) proposed by Ren et al. (2019) defines the efficiency of a DST model using J, the number of slots, and M, the number of values of a slot. Going a step further from their work, we report ITC of the models in the best case and the worst case for relatively more precise comparison.

Table 11 shows ITC of several models in their best and worst cases. Since our model generates values for only the slots on which **UPDATE** operation has to be performed, the best case complexity of our model is $\Omega(1)$, when there is no slot whose operation is **UPDATE**.

C Sample Outputs

Turn	Dialogue	Pred B_{t-1} (Model Output at Turn $t-1$)	Pred B_t (Model Output at Turn t)	Ground Truth B_t
1	U: i am looking for a cheap restaurant in the center of the city .		restaurant-area-centre restaurant-pricerange-cheap	restaurant-area-centre restaurant-pricerange-cheap
2	A: do you have any specific type of food you would like ? U: no, i am not picky as long as the price –s are low.	restaurant-area-centre restaurant-pricerange-cheap	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare	restaurant-area-centre restaurant-pricerange-cheap
3	A: there is a cheap chinese restaurant called the dojo noodle bar located in the centre of town. would you like to book a table ? U: yes please, for 8 people at 18:30 on thursday .	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-18:30 restaurant-book people-8 restaurant-book day-thursday	restaurant-area-centre restaurant-pricerange-cheap restaurant-book time-18:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar
4	A: i am sorry but dojo noodle bar is solidly booked at that time . i can try a different time or day for you. U: can you try to book it at 17:30 .	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-18:30 restaurant-book people-8 restaurant-book day-thursday	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar	restaurant-area-centre restaurant-pricerange-cheap restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar
5	A: all set . your reference number is k2bo09vq . U: thanks . i am also looking for some entertainment close to the restaurant . any suggestions ?	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-entertainment	restaurant-area-centre restaurant-pricerange-cheap restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-entertainment
6	A: i am sorry I do not have any attractions meeting the criteria you listed . is there another type of attraction or area you would like me to search ? U: why do not you try an architectural attraction, instead .	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-entertainment	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture	restaurant-area-centre restaurant-pricerange-cheap restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday Rrstaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture
7	A: all saints church looks good , would you like to head there ? U: yes please, can you let me know what the entrance fee is too ?	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture attraction-name-all saints church	restaurant-area-centre restaurant-pricerange-cheap restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture attraction-name-all saints church
8	A: its entrance fee is free . U: i also need to book a taxi between the restaurant and the church .	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture attraction-name-all saints church	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture attraction-name-all saints church taxi-departure-dojo noodle bar	restaurant-area-centre restaurant-pricerange-cheap restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture attraction-name-all saints church taxi-departure-dojo noodle bar taxi-destination-all saints church
9	A: what time would you like the taxi to pick you up from dojo noodle bar ? U: 18:30 , please .	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture attraction-name-all saints church taxi-departure-dojo noodle bar	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture attraction-name-all saints church taxi-departure-dojo noodle bar taxi-leaveat-18:30	restaurant-area-centre restaurant-pricerange-cheap restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture attraction-name-all saints church taxi-departure-dojo noodle bar taxi-destination-all saints church taxi-leaveat-18:30
10	A: your taxi is booked . the car is a blue bmw and the contact number is 07023944669 . is there anything else I can help you with ? U: that s it . thanks .	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture attraction-name-all saints church taxi-departure-dojo noodle bar taxi-leaveat-18:30	restaurant-area-centre restaurant-pricerange-cheap restaurant-food-dontcare restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture attraction-name-all saints church taxi-departure-dojo noodle bar taxi-leaveat-18:30	restaurant-area-centre restaurant-pricerange-cheap restaurant-book time-17:30 restaurant-book people-8 restaurant-book day-thursday restaurant-name-dojo noodle bar attraction-area-centre attraction-type-architecture attraction-name-all saints church taxi-departure-dojo noodle bar taxi-destination-all saints church taxi-leaveat-18:30

Figure 3: The output of SOM-DST in a dialogue (`dialogue_idx` MUL2499) in the test set of MultiWOZ 2.1. Parts changed from the previous dialogue state are shown in blue. To save space, we omit the slots with value `NULL` from the figure.

Turn	Dialogue	Pred B_{t-1} (Model Output at Turn $t-1$)	Pred B_t (Model Output at Turn t)	Ground Truth B_t
1	U: i am planing a trip to cambridge and i need ideas on a place i can go in the centre		attraction-area-centre	attraction-area-centre
2	A: there is much to choose from in the centre . there are several architecture attractions to explore , boats and swimming , colleges , museums , parks , theatres and even nightclubs . U: what do you recommend ?	attraction-area-centre	attraction-area-centre	attraction-area-centre
3	A: i recommend whipple museum of the history of science . it s educational and free ! U: okay that sounds fun .	attraction-area-centre	attraction-area-centre attraction-name-whipple museum of the history of science	attraction-area-centre attraction-name-whipple museum of the history of science
4	A: is there anything else i can help you with today ? U: what can you tell me about the express by holiday inn cambridge ?	attraction-area-centre attraction-name-whipple museum of the history of science	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge
5	A: sure it is located in the east and is in the expensive price range and their telephone number is 01223866800 . U: can you book it for me please ? i need it for 1 person and 5 nights starting on tuesday .	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge hotel-book day-5 hotel-book people-1 hotel-book day-thuesday	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge hotel-book day-5 hotel-book people-1 hotel-book day-thuesday
6	A: i was able to book your stay at the express by holiday inn cambridge . your reference number is ye0m6sg1 . U: great ! thanks . i was also hoping to book a taxi , can you help me with that ?	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge hotel-book day-5 hotel-book people-1 hotel-book day-thuesday	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge hotel-book day-5 hotel-book people-1 hotel-book day-thuesday	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge hotel-book day-5 hotel-book people-1 hotel-book day-thuesday
7	A: yes i can , what is the departure and destionation site , and when would you like to leave and arrive by ? U: i would like to leave the hotel by 02:45 to go to the museum .	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge hotel-book day-5 hotel-book people-1 hotel-book day-thuesday	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge hotel-book day-5 hotel-book people-1 hotel-book day-thuesday taxi-departure-express by holiday inn cambridge taxi-destination-whipple-museum of the history of science taxi-leaveat-02:45	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge hotel-book day-5 hotel-book people-1 hotel-book day-thuesday taxi-departure-express by holiday inn cambridge taxi-destination-whipple-museum of the history of science taxi-leaveat-02:45
8	A: okay , i have got that booked for you . you are taxi is a yellow honda and the contact number is 07272096370 . can i do anything else today ? U: thanks alot for helping	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge hotel-book day-5 hotel-book people-1 hotel-book day-thuesday taxi-departure-express by holiday inn cambridge taxi-destination-whipple-museum of the history of science taxi-leaveat-02:45	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge hotel-book day-5 hotel-book people-1 hotel-book day-thuesday taxi-departure-express by holiday inn cambridge taxi-destination-whipple-museum of the history of science taxi-leaveat-02:45	attraction-area-centre attraction-name-whipple museum of the history of science hotel-name-express by holiday inn cambridge hotel-book day-5 hotel-book people-1 hotel-book day-thuesday taxi-departure-express by holiday inn cambridge taxi-destination-whipple-museum of the history of science taxi-leaveat-02:45

Figure 4: The output of SOM-DST in a dialogue (`dialogue_idx` PMUL3748) in the test set of MultiWOZ 2.1. Parts changed from the previous dialogue state are shown in blue. To save space, we omit the slots with value `NULL` from the figure.