

Find or Classify? Dual Strategy for Slot-Value Predictions on Multi-Domain Dialog State Tracking

Jian-Guo Zhang^{1*} Kazuma Hashimoto^{2†} Chien-Sheng Wu² Yao Wan³

Philip S. Yu¹ Richard Socher² Caiming Xiong²

¹ University of Illinois at Chicago, Chicago, USA

² Salesforce Research, Palo Alto, USA

³ Huazhong University of Science and Technology, Wuhan, China

{jzhan51, psyu}@uic.edu, wanyao@hust.edu.cn

{k.hashimoto, wu.jason, rsocher, cxiong}@salesforce.com

Abstract

Dialog state tracking (DST) is a core component in task-oriented dialog systems. Existing approaches for DST mainly fall into one of two categories, namely, ontology-based and ontology-free methods. **An ontology-based method selects a value from a candidate-value list for each target slot, while an ontology-free method extracts spans from dialog contexts.** Recent work introduced a BERT-based model to strike a balance between the two methods by pre-defining categorical and non-categorical slots. **However, it is not clear enough which slots are better handled by either of the two slot types, and the way to use the pre-trained model has not been well investigated.** In this paper, we propose a simple yet effective dual-strategy model for DST, by adapting a single BERT-style reading comprehension model to jointly handle both the categorical and non-categorical slots. **Our experiments on the MultiWOZ datasets show that our method significantly outperforms the BERT-based counterpart, finding that the key is a deep interaction between the domain-slot and context information.** When evaluated on noisy (MultiWOZ 2.0) and cleaner (MultiWOZ 2.1) settings, our method performs competitively and robustly across the two different settings. Our method sets the new state of the art in the noisy setting, while performing more robustly than the best model in the cleaner setting. We also conduct a comprehensive error analysis on the dataset, including the effects of the dual strategy for each slot, to facilitate future research.

1 Introduction

Virtual assistants play important roles in facilitating our daily life, such as booking hotels, reserving restaurants and making travel plans. Dialog State

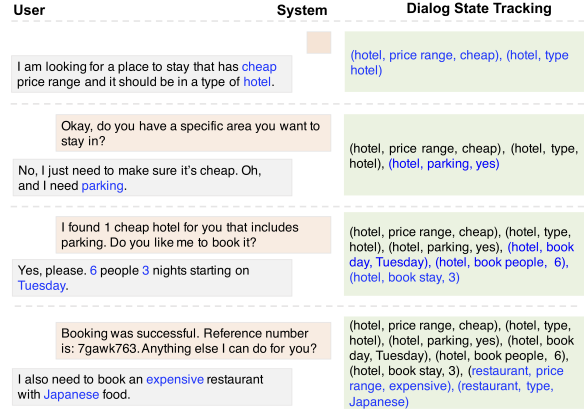


Figure 1: An example of dialog state tracking for booking a hotel and reserving a restaurant. Each turn contains a user utterance (grey) and a system utterance (orange). The dialog state tracker (green) tracks all the $\langle domain, slot, value \rangle$ triplets until the current turn. Blue color denotes the new state appearing at that turn. Best viewed in color.

Tracking (DST), which estimates users' goal and intention based on conversation history, is a core component in task-oriented dialog systems (Young et al., 2013; Gao et al., 2019a). A dialog state consists of a set of $\langle domain, slot, value \rangle$ triplets, and DST aims to track all the states accumulated across the conversational turns. Fig. 1 shows a dialogue with corresponding annotated turn states.

Traditional approaches for DST usually rely on hand-crafted features and domain-specific lexicon, and can be categorized into two classes (Xu and Hu, 2018; Gao et al., 2019b; Ramadan et al., 2018; Zhong et al., 2018): i.e., ontology-based and ontology-free. The ontology-based approaches (Ramadan et al., 2018; Zhong et al., 2018; Chen et al., 2020) require full access to the pre-defined ontology to perform classification over the candidate-value list. However, in practice, we may not have access to an ontology or only have partial ontol-

*Work done while the first author was an intern at Salesforce Research.

† Corresponding author.

ogy in the industry. Even if a full ontology exists, it is computationally expensive to enumerate all the values when the full ontology for some slots is very large and diverse (Wu et al., 2019; Xu and Hu, 2018). The ontology-free approaches (Gao et al., 2019b; Xu and Hu, 2018) find slot values directly from the input source using a copy mechanism without requiring an ontology, e.g., learning span matching with start and end positions in the dialog context. However, it is nontrivial to handle situations where values do not appear in the dialog context or have various descriptions by users.

To mitigate the above issues, recently, (Zhou and Small, 2019) introduced a question asking model to generate questions asking for values of each-domain slot pair and a dynamic knowledge graph to learn relationships between the (domain, slot) pairs. (Rastogi et al., 2020) introduced a BERT-based model (Devlin et al., 2019) to strike a balance between the two methods by pre-defining categorical and non-categorical slots. However, more studies are needed to know which slots are better handled by either of the two slot types, and the way to use the pre-trained models is not well investigated (Lee et al., 2019; Gao et al., 2019b; Rastogi et al., 2020).

Inspired by the task-oriented dialog schema design in (Rastogi et al., 2020) and the recent successful experience in locating text spans in machine reading comprehensions (Gao et al., 2019b; Asai et al., 2019), we design a simple yet effective **Dual-Strategy Dialog State Tracking** model (**DS-DST**), which adapts a single BERT question answering model to jointly handle both the categorical and non-categorical slots, and different with previous approaches on multi-domain DST, we enable the model with direct interactions between dialog context and the slot. We decide whether a slot belongs to a non-categorical slot or a categorical slot by following the heuristics from (Rastogi et al., 2020). For example, it is common that when users book hotels, the requests for parking are usually *yes* or *no* with limited choices. These kinds of slots are defined as categorical slots, and the slot values are selected over a partial ontology. In addition, how long the user will stay has unlimited values and it can be found in the context. These kinds of slots are treated as non-categorical slots, and the values are found through span matching in the dialog context. Hence, the model is flexible depending on the access level to the ontology or whether the values of slots could be found directly in the dialog

context.

Our contributions are summarized as follows:

- We designed a simple yet effective dual-strategy model based on BERT with strong interactions between the dialog context and domain-slot pairs.
- Our model achieves state of the art on MultiWOZ 2.0 (Budzianowski et al., 2018) and competitive performance on MultiWOZ 2.1 (Eric et al., 2019). Our model also performs robustly across the two different settings.
- We conducted a comprehensive error analysis on the dataset, including the effects of the dual strategy for each slot, to facilitate future research.

2 Related Work

Multi-domain DST, which tracks dialog states in complicated conversations across multiple domains with many slots, has been a hot research topic during the past few years, along with the development of Dialogue State Tracking Challenges (Williams et al., 2013; Henderson et al., 2014a,b; Kim et al., 2016, 2017, 2019). Traditional approaches usually rely on hand-crafted features or domain-specific lexicon (Henderson et al., 2014c; Wen et al., 2016), making them difficult to be adapted to new domains. In addition, these approaches require a pre-defined full ontology, in which the values of a slot are constrained by a set of candidate values (Ramadan et al., 2018; Liu and Lane, 2017; Zhong et al., 2018; Lee et al., 2019; Chen et al., 2020). To tackle these issues, several methods have been proposed to extract slot values through span matching with start and end positions in the dialog context. For example, (Xu and Hu, 2018) utilizes an attention-based pointer network to copy values from the dialog context. (Gao et al., 2019b) poses DST as a reading comprehension problem and incorporates a slot carryover model to copy states from previous conversational turns. However, tracking states only from the dialog context is insufficient since many values in DST cannot be exactly found in the context due to annotation errors or diverse descriptions of slot values from users. On the other hand, pre-trained models such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2018) have shown promising performances in many downstream tasks. Among them, DSTreader (Gao et al., 2019b) utilizes BERT as word embeddings for dialog contexts, SUMBT (Lee et al., 2019) employs BERT to extract representations of candidate values, and

BERT-DST (Rastogi et al., 2020) adopts BERT to encode the inputs of the user turn as well as the previous system turn. Different from these approaches where the dialog context and domain-slot pairs are usually separately encoded, we employ strong interactions to encode them.¹ Moreover, We investigate and provide insights to decide slot types and conduct a comprehensive analysis of the popular MultiWOZ datasets.

Another direction for multi-domain DST is based on generative approaches (Lei et al., 2018; Wu et al., 2019; Le et al., 2020) which generate slot values without relying on fixed vocabularies and spans. However, such generative methods suffer from generating ill-formatted strings (e.g., repeated words) upon long strings, which is common in DST. For example, the hotel address may be long and a small difference makes the whole dialog state tracking incorrect. By contrast, both the categorical (picklist-based) and non-categorical (span-based) methods can rely on existing strings rather than generating them.

3 DS-DST: a Dual Strategy for DST

Let $X = \{(U_1^{sys}, U_1^{usr}), \dots, (U_T^{sys}, U_T^{usr})\}$ denote a set of pairs of a system utterance U_t^{sys} and a user utterance U_t^{usr} ($1 \leq t \leq T$), given a dialogue context with T turns. Each turn (U_t^{sys}, U_t^{usr}) talks about a particular domain (e.g., *hotel*), and a certain number of slots (e.g., *price range*) are associated with the domain. We denote all the N possible domain-slot pairs as $S = \{S_1, \dots, S_N\}$, where each domain-slot pair consists of $\{s_1, \dots, s_n\}$ tokens, e.g., *hotel-price range* includes three tokens. Let $X_t = \{(U_1^{sys}, U_1^{usr}), \dots, (U_t^{sys}, U_t^{usr})\}$ denote the dialogue context at the t_{th} turn and X_t has $\{x_1, \dots, x_m\}$ tokens. Our goal is to predict the values for all the domain-slot pairs in S . Here we assume that M domain-slot pairs in S are treated as non-categorical slots, and the remaining $N - M$ pairs as categorical slots. Each categorical slot has L possible candidate values (picklist), i.e., $\{V_1, \dots, V_L\}$, where L is the size of the picklist, and each value has $\{v_1, \dots, v_c\}$ tokens.

Bearing these notations in mind, we then propose a dual strategy model with direct interactions between dialog context and domain-slot pairs for DST. Fig. 2 shows an overview of the architecture

¹Recent work on question answering has shown that the joint encoding of query-context pairs is crucial to achieving high accuracy (Qiu et al., 2019; Asai et al., 2019)

of our proposed DS-DST model. We first utilize a pre-trained BERT (Devlin et al., 2019) to encode information about the dialogue context X_t along with each domain-slot pair in S , and obtain contextualized representations conditioned on the domain-slot information. We then design a slot gate to handle special types of values. In particular, for the non-categorical slots, we utilize a two-way linear mapping to find text spans. For the categorical slots, we select the most plausible values from the picklists based on the contextual representation.

3.1 Slot-Context Encoder

We employ a pre-trained BERT (Devlin et al., 2019) to encode the domain-slot types and dialog contexts. For the j_{th} domain-slot pair and the dialog context X_t at the t_{th} turn, we concatenate them and get corresponding representations:

$$R_{tj} = \text{BERT}([\text{CLS}] \oplus S_j \oplus [\text{SEP}] \oplus X_t), \quad (1)$$

where $[\text{CLS}]$ is a special token added in front of each sample, and $[\text{SEP}]$ is a special separator token. The outputs of BERT in Eq. (3.1) can be decomposed as $R_{tj} = [r_{tj}^{\text{CLS}}, r_{tj}^1, \dots, r_{tj}^K]$, where r_{tj}^{CLS} is the aggregated representation of the total K sequential input tokens, and $[r_{tj}^1, \dots, r_{tj}^K]$ are the token-level representations. They are used for slot-value predictions in the following sections, and the BERT is fine-tuned during the training process.

3.2 Slot-Gate Classification

As there are many domain-slot pairs in multi-domain dialogues, it is nontrivial to correctly predict whether a domain-slot pair appears at each turn of the dialogue. Here we follow (Wu et al., 2019; Xu and Hu, 2018) and design a slot gate classification module for our neural network. Specifically, at the t_{th} turn, the classifier makes a decision among $\{\text{none}, \text{dontcare}, \text{prediction}\}$, where *none* denotes that a domain-slot pair is not mentioned or the value is ‘none’ at this turn, *dontcare* implies that the user can accept any values for this slot, and *prediction* represents that the slot should be processed by the model with a real value. We utilize r_{tj}^{CLS} for the slot-gate classification, and the probability for the j_{th} domain-slot pair at the t_{th} turn is calculated as:

$$P_{tj}^{\text{gate}} = \text{softmax}(W_{\text{gate}} \cdot (r_{tj}^{\text{CLS}})^\top + b_{\text{gate}}), \quad (2)$$

where W_{gate} and b_{gate} are learnable parameters and bias, respectively.

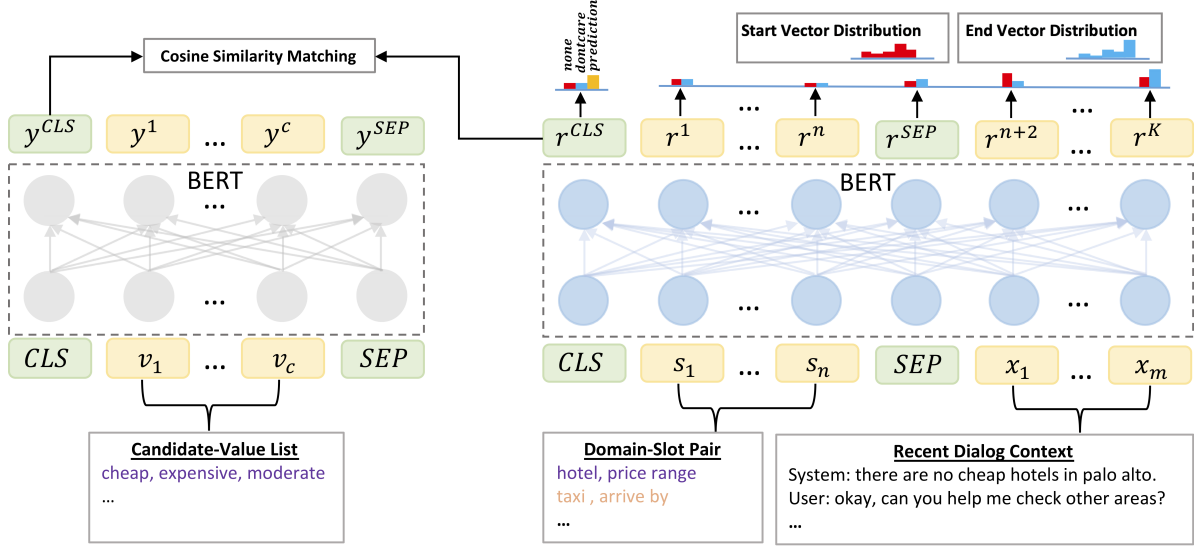


Figure 2: The architecture of our proposed DS-DST model. The left part is a fixed BERT model which acts as a feature extractor and outputs the representations of values in the candidate-value list for each categorical slot (marked in purple). The right part is the other fine-tuned BERT model which outputs representations for the concatenation of each domain-slot pair and the recent dialog context.

We adopt the cross-entropy loss function for the slot gate classification as follows:

$$\mathcal{L}_{gate} = \sum_{t=1}^T \sum_{j=1}^N -\log(P_{tj}^{gate} \cdot (y_{tj}^{gate})^\top), \quad (3)$$

where y_{tj}^{gate} is the one-hot gate label for the j_{th} domain-slot pair at the t_{th} turn.

3.3 Non-Categorical Slot-Value Prediction

For each non-categorical slot, its value can be mapped to a span with start and end position in the dialog context, e.g., slot *leave at* in the *taxi* domain has spans 4 : 30pm in the context. We take token-level representations $[r_{tj}^1, \dots, r_{tj}^K]$ of the dialog context as input, and apply a two-way linear mapping to get a start vector α_{tj}^{start} and an end vector α_{tj}^{end} :

$$\begin{bmatrix} \alpha_{tj}^{start} \\ \alpha_{tj}^{end} \end{bmatrix} = W_{span} \cdot ([r_{tj}^1, \dots, r_{tj}^K])^\top + b_{span}, \quad (4)$$

where W_{span} and b_{span} are learnable parameters and bias, respectively.

The probability of the i_{th} word being the start position of the span is computed as: $p_{tj}^{start_i} = \frac{e^{\alpha_{tj}^{start} \cdot r_{tj}^i}}{\sum_k e^{\alpha_{tj}^{start} \cdot r_{tj}^k}}$, and the loss for the start position prediction can be calculated as:

$$\mathcal{L}_{start} = \sum_{t=1}^T \sum_{j=1}^M -\log(P_{tj}^{start} \cdot (y_{tj}^{start})^\top), \quad (5)$$

where y_{tj}^{start} is the one-hot start position label for the j_{th} domain-slot pair at the t_{th} turn.

Similarly, we can also get the loss \mathcal{L}_{end} for end positions prediction. Therefore, the total loss \mathcal{L}_{span} for the non-categorical slot-value prediction is the summation of \mathcal{L}_{start} and \mathcal{L}_{end} .

3.4 Categorical Slot-Value Prediction

Each categorical slot has several candidate values; e.g., slot *price range* in the *hotel* domain has three values $\{cheap, expensive, moderate\}$. At the t_{th} turn, for the j_{th} domain-slot pair, we first use another pre-trained BERT to get the aggregated representation of each value in the candidate list:

$$y_l^{CLS} = \text{BERT}([\text{CLS}] \oplus V_l \oplus [\text{SEP}]), \quad (6)$$

where $l \in \{1, \dots, L\}$. Note that during the training process, this separate BERT model acts as a feature extractor and its model parameters are fixed.

We calculate the relevance score between the aggregated representation and a reference candidate by the cosine similarity (Lin et al., 2017):

$$\cos(r_{tj}^{CLS}, y_l^{CLS}) = \frac{r_{tj}^{CLS} \cdot (y_l^{CLS})^\top}{\|r_{tj}^{CLS}\| \|y_l^{CLS}\|}, \quad (7)$$

where r_{tj}^{CLS} and y_l^{CLS} are the aggregated representations from the slot-context encoder and the reference candidate value, respectively.

During the training process, we employ a hinge loss to enlarge the difference between the similarity

Domain	Hotel	Train	Restaurant	Attraction	Taxi
Slots	price range type parking book stay book day book people area stars internet name	destination day departure arrive by book people leave at	food price range area name book time book day book people	area name type	leave at destination departure arrive by
Train	3381	3103	3813	2717	1654
Validation	416	484	438	401	207
Test	394	494	437	395	195

Table 1: The dataset information of MultiWOZ 2.0 and MultiWOZ 2.1. The top two rows list 5 selected domains, consisting of 30 domain-slot pairs. The last three rows show the number of dialogues for each domain.

of r_{tj}^{CLS} to the target value and that to the most similar value in the candidate-value list:

$$\mathcal{L}_{picklist} = \sum_{t=1}^T \sum_{j=1}^{N-M} \max(0, \lambda - \cos(r_{tj}^{\text{CLS}}, y_{target}^{\text{CLS}})) + \max_{y_l^{\text{CLS}} \neq y_{target}^{\text{CLS}}} \cos(r_{tj}^{\text{CLS}}, y_l^{\text{CLS}}), \quad (8)$$

where λ is a constant margin and $l \in \{1, \dots, L\}$, and $\mathcal{L}_{picklist}$ only requires partial ontology for DS-DST.

3.5 Training Objective

During training process, the above three modules can be jointly trained and share parameters of BERT. We optimize the summations of different losses as:

$$\mathcal{L}_{total} = \mathcal{L}_{gate} + \mathcal{L}_{span} + \mathcal{L}_{picklist}. \quad (9)$$

For the slots that are not mentioned or the users can accept any values (i.e., slots $\in \{none, dontcare\}$) at each dialogue turn, \mathcal{L}_{span} and $\mathcal{L}_{picklist}$ are set to 0 and only the slot-gate classification is optimized during the training process.

4 Experimental Setup

4.1 Datasets

We use the MultiWOZ 2.0 (Budzianowski et al., 2018) dataset and the MultiWOZ 2.1 (Eric et al., 2019) dataset. MultiWOZ 2.0 is one of the largest multi-domain dialogue corpora with seven distinct domains and over 10,000 dialogues. Compared with the original dataset, MultiWOZ 2.1 conducts dataset correction, including correcting dialog states, spelling errors, dialogue utterance corrections, and mis-annotations to reduce several sub-

stantial noises (more details can be found in Eric et al. (2019)).

As *hospital* and *police* domains contain very few dialogues (5% of total dialogues), and they only appear in the training dataset, we ignore them in our experiments, following Wu et al. (2019). We adopt only five domains (i.e., *train*, *restaurant*, *hotel*, *taxi*, *attraction*) and obtain totally 30 domain-slot pairs in the experiments. Table 1 summarizes the domain-slot pairs and their corresponding statistics in MultiWOZ 2.0 and MultiWOZ 2.1. We follow the standard training/validation/test split strategy provided in the original datasets, and the data pre-processing script provided in Wu et al. (2019).

For MultiWOZ 2.0 and 2.1, the candidate values for the categorical slots are derived based on the ground-truth values of each slot that appeared in the partial dataset ontology. Besides, Since there are no provided ground-truth start positions and end positions for non-categorical slots in the datasets, we find the spans through string matching between the ground truth values and the values in the dialog contexts, and we treat the start and end positions of the span which appeared at the most recent dialog turn as the ground-truth start positions and end positions.

4.2 Models

We make a comparison with several existing models² and introduce some of them as below:

- **SpanPtr** (Xu and Hu, 2018). It applies a RNN-based pointer network to find text spans with start and end pointers for each domain-slot pair.
- **Ptr-DST**. It is a variant based on SpanPtr with the exception that some slots are categorical slots, following DS-DST.
- **DSTreader** (Gao et al., 2019b). It models the DST from the perspective of machine reading comprehensions and applies a pre-trained BERT as initial word embeddings.
- **DSTQA** (Zhou and Small, 2019). It applies a dynamically-evolving knowledge graph and generates question asking for the values of a domain-slot pair.
- **TRADE** (Wu et al., 2019). It contains a slot gate module for slots classification and a pointer generator for states generation.
- **COMER** (Ren et al., 2019). It applies BERT as contextualized word embeddings and first gen-

²We did not compare with (Lee et al., 2019) and (Shan et al., 2020) as the datasets preprocessing is different with other baselines and ours.

Models	MultiWOZ 2.0	MultiWOZ 2.1
SpanPtr (Xu and Hu, 2018)	30.28%	29.09%
Ptr-DST	-	42.17%
DSTreader (Gao et al., 2019b)	39.41%	36.40%*
TRADE (Wu et al., 2019)	48.62%	45.60%*
COMER (Ren et al., 2019)	45.72%	-
DSTQA w/span (Zhou and Small, 2019)	51.36%	49.67%
DSTQA w/o span ⁺ (Zhou and Small, 2019)	51.44%	51.17%
BERT-DST (Rastogi et al., 2020)	-	43.40%
MA-DST (Kumar et al., 2020)	-	51.04%
SST-2 ⁺ (Chen et al., 2020)	51.17%	55.23%
NA-DST (Le et al., 2020)	50.52%	49.04%
DS-Span	42.59%	40.00%
DS-DST	52.24%	51.21%
DS-Picklist ⁺	54.39%	53.30%

Table 2: Joint accuracy on the test sets of MultiWOZ 2.0 and 2.1. ⁺: the models require a full ontology, and *: the results are reported by Eric et al. (2019)

erates the slot sequences in the belief state, then generates the value sequences for each slot.

- **BERT-DST** (Rastogi et al., 2020). It uses BERT to obtain schema element embeddings and encode system as well as user utterances for dialogue state tracking. Different from the original model, it incorporates a pointer-generator copying mechanism for non-categorical slots of the MultiWOZ datasets.

- **SST-2** (Chen et al., 2020). It uses the graph neural network to incorporate slot relations and model slot interactions.

For our proposed methods, we design three variants:

- **DS-DST**. It represents our proposed dual strategy model for DST, which can simultaneously handle the non-categorical slots as well as the categorical ones. Following heuristics from (Rastogi et al., 2020), *time* and *number* related slots are treated as non-categorical slots, resulting in five slot types across four domains (nine domain-slot pairs in total), and the rest slots are treated as categorical slots (See also in Table 4). We also conduct investigations to decide domain-slot types in the experiments.

- **DS-Span**. Similar to Xu and Hu (2018); Gao et al. (2019b), it treats all domain-slot pairs as non-categorical slots, where corresponding values for each slot are extracted through text spans (string matching) with start and end positions in the dialog context.

- **DS-Picklist**. Similar to (Lee et al., 2019; Chen et al., 2020), It assumes a full ontology is available

Models	Joint Accuracy
BERT-DST (Rastogi et al., 2020)	43.40%
DS-DST	51.21%
BERT-DST-Picklist (single turn)	39.86%
BERT-DST-Picklist (whole dialog history)	46.42%
ToD-BERT (Wu et al., 2020)	48.00%
DS-Picklist	53.30%

Table 3: Joint accuracy on the test sets of MultiWOZ 2.1. BERT-DST is the model used in MultiWOZ 2.1. BERT-DST-Picklist is the original model described in (Rastogi et al., 2020), where a full ontology is required and all the slots are treated as categorical slots,. ‘single turn’ and ‘whole dialog history’ represent the Bert utterance inputs are the current dialog turn and the whole dialog history, respectively.

and treats all domain-slot pairs as categorical slots, where corresponding values for each slot are found in the candidate-value list (picklist).

5 Experimental Results

We evaluate all the models using the joint accuracy metric. At each turn, the joint accuracy is 1.0 if and only if all $\langle domain, slot, value \rangle$ triplets are predicted correctly, otherwise 0. The score is averaged across all the turns in the evaluation set.

5.1 Joint Accuracy

Overall performance Table 2 shows the results on the test sets of two datasets. We can see that our models achieve the top performance on MultiWOZ 2.0 and competitive performance on MultiWOZ 2.1. Among these state-of-the-art results, ours are less sensitive to the dataset differences.

Slot Name	DS-Span	DS-DST	DS-Picklist
hotel-type	87.92	93.97 (+6.05)	94.29 (+6.37)
attraction-name	91.16	93.81 (+2.65)	93.93 (+2.77)
restaurant-name	92.11	93.38 (+1.27)	92.89 (+0.78)
hotel-internet	92.98	97.48 (+4.50)	97.26 (+4.28)
hotel-parking	93.42	97.18 (+3.76)	96.99 (+3.57)
attraction-type	93.77	96.86 (+3.09)	96.91 (+3.14)
hotel-name	94.19	94.87 (+0.68)	94.77 (+0.58)
hotel-area	94.73	95.87 (+1.14)	95.47 (+0.74)
restaurant-area	96.23	96.86 (+0.63)	97.18 (+0.95)
attraction-area	96.57	96.96 (+0.39)	96.73 (+0.16)
hotel-price range	96.92	97.39 (+0.47)	96.97 (+0.05)
train-departure	96.96	98.55 (+1.59)	98.34 (+1.38)
restaurant-food	97.24	97.60 (+0.36)	97.19 (-0.05)
restaurant-price range	97.29	97.73 (+0.44)	97.69 (+0.40)
taxi-departure	97.57	98.53 (+0.96)	98.59 (+1.02)
taxi-destination	97.69	98.49 (+0.80)	98.24 (+0.55)
hotel-stars	97.80	97.48 (-0.32)	97.76 (-0.04)
train-destination	98.17	98.86 (+0.69)	98.59 (+0.42)
train-day	99.24	99.35 (+0.11)	99.33 (+0.09)
hotel-book day	99.40	99.32 (-0.08)	99.24 (-0.16)
restaurant-book day	99.40	99.57 (+0.17)	99.44 (+0.04)
train-leave at	93.43	93.30 (-0.13)	93.91 (+0.48)
train-arrive by	95.25	95.78 (+0.53)	96.59 (+1.34)
train-book people	97.99	97.84 (-0.15)	98.51 (+0.52)
restaurant-book time	98.56	98.44 (-0.12)	99.04 (+0.48)
taxi-leave at	98.63	98.53 (-0.10)	98.94 (+0.31)
hotel-book people	99.06	99.04 (-0.02)	99.29 (+0.23)
taxi-arrive by	99.12	99.01 (-0.11)	99.09 (-0.03)
hotel-book stay	99.25	99.25 (+0.00)	99.40 (+0.15)
restaurant-book people	99.31	99.16 (-0.15)	99.44 (+0.13)
Average Accuracy	96.38%	97.35%	97.40%

Table 4: The slot-level accuracy on the test set of MultiWOZ 2.1. ‘+/-’ indicates absolute performance improvement/degradation compared with DS-Span. The numbers highlighted in bold indicate that the difference is significant ($p < 0.05$), tested by bootstrap resampling (Noreen, 1989). The slots above the first dashed line are categorical slots and the slots below the first dashed line are non-categorical slots for DS-DST. The last row shows the average slot accuracy.

Comparing DS-Span and DS-DST, we can find that jointly using the non-categorical and categorical approaches is indeed helpful in multi-domain DST. When the model has access to the full ontology, DS-Picklist shows that our method could further improve the DST performance. Although DS-Picklist is higher than DS-DST, in real scenarios, it may be nontrivial to have access to the full ontology. In the paper, we jointly train the three modules in Section 3.5, we also conduct experiments for separately training the non-categorical slots and categorical slots. DS-DST drops by 1.90% on MultiWOZ 2.1, which shows the benefits of jointly training.

Detailed comparisons with BERT related methods Compared with those methods as shown in Table 2, we can observe that DS-Span, which employs the strength of BERT, outperforms SpanPtr

by 10.91%, and it outperforms COMMER and DSTreader, which also use a pre-trained BERT model as dialog context embeddings and word embeddings, respectively. DS-DST outperforms BERT-DST, which separately encodes dialog context and domain-slot pairs based on BERT, by 7.81% on MultiWOZ 2.1. The above results shows the effectiveness of our model design based on BERT, where we enforce the strong interactions between dialog context and domain-slot pairs.

To further investigate the differences and importance of strong interactions, we reproduce the original BERT-DST model described in (Rastogi et al., 2020). In addition, we compare with ToD-BERT (Wu et al., 2020), which is a large pre-trained model based on several task-oriented dialogue datasets, and it also separately encodes dialog context and domain-slot pairs. We show the results in Table 3.³ We observe that our model is consistently much better than BERT-DST and BERT-DST-Picklist. Moreover, our models based on BERT surpass the strong ToD-BERT. We conclude that our improvements come from the strong interactions between slots and dialog context. Therefore, it is important to employ strong interactions to multi-domain DST tasks.

5.2 Per Slot Accuracy

Now that we have observed that DS-DST and DS-Picklist perform much better than DS-Span, we focus on where the accuracy improvement comes from. Table 4 shows the accuracy for each slot type on the MultiWOZ 2.1 test set, and we can observe significant improvement over the DS-Span baseline for some slots, including *hotel-type*, *attraction-type*, *attraction-name*, *hotel-internet* and *hotel-parking*. This is because their values usually have different expressions and cannot be extracted from the dialog context, which decreases the performance of the span-based methods. In contrast, their values can be predicted directly from the candidate-value lists. Compared with other slots, these slots still have space for improvements.

5.3 Analysis and Discussions

Error analysis To better understand the improvement, we conducted an error analysis and inspected actual examples on the MultiWOZ 2.1 validation set. Table 5 shows the top-10 slots, according to

³Here we did not show the results when treating all the slots as non-categorical slots, one reason is that the performances of BERT-DST-Span are much worse than BERT-DST.

Slot Name	DS-Span (#Unfound / #Relative_Turns)	DS-DST	DS-Picklist
hotel-type	667/1395	86.36%	85.91%
hotel-parking	419/1048	89.50%	86.63%
hotel-internet	421/1124	95.72%	94.54%
taxi-leave at	73/364	0.00%	43.84%
attraction-name	215/1261	70.23%	74.42%
attraction-type	270/1658	84.81%	84.07%
train-leave at	181/1164	2.21%	41.44%
hotel-area	168/1452	51.19%	58.93%
train-arrive by	125/1428	9.60%	79.20%
attraction-area	177/1620	67.23%	71.75%

Table 5: Statistics of Top-10 slots on the MultiWOZ 2.1 validation set based on (#Unfound / #Relative_Turns). DS-DST and DS-Picklist show percentages based on (#Recover / #Unfound). #Unfound is the number of slots whose values cannot be found through span matching in the dialog context, #Relative_Turns is the number of dialogue turns where the slot type is mentioned, and #Recover indicates the number of values correctly predicted by DS-DST or DS-Picklist.

the ratio of ground-truth slot values which cannot be found through span matching. That is, for such examples, DS-Span cannot extract the ground-truth strings, resulting in the low joint accuracy. Here, we show how well our DS-DST and DS-Picklist can correctly predict the missing values in DS-Span. As we can see in this table, the two methods dramatically reduce the errors for some slots such as *attraction-type*, *hotel-internet* and *hotel-parking*. Hence, for these kinds of slots, it is better to treat them as categorical slots. Among the top-10 slots, the *time*-related slots such as *taxi-leave at* and *train-arrive by*, which are span-based slots in DS-DST, DS-Span and DS-DST cannot perform well as there are no span matching in the dialogue context, and only few values (i.e., ‘none’ and ‘dontcare’) can be correctly predicted by the slot-gate classification. When the ontology is accessible, DS-Picklist can further reduce the error rates, since the predicted values can be found in the candidate-values lists.

On the other hand, we also investigated slots whose ground-truth values can be found through span matching, and we did not observe a significant difference between the three methods. This means that both the non-categorical and categorical methods perform similarly when target values are explicitly mentioned in the dialogues. Therefore, when most of the slot values can be found directly in the dialog context, these slots can be treated as either non-categorical slots or categorical slots.

As our model relies on the slot-gate classification in Section 3.2, we also investigate the potential influence of this module. We replace this mod-

ule with an oracle slot-gate classification module, and the joint accuracy is improved from 55.23% to 86.10% on the development set of MultiWOZ 2.1, which indicates that there is a great space to improve the performance with better designs of the slot-gate classification module.

Examples Table 6 shows three examples of dialogue turns in the validation set. In the first example, we can see that DS-Span cannot correctly extract the ground-truth values, because the User does not always explicitly mention ‘yes’ or ‘no’ when being asked about the internet or parking requests. In the second example, the User and the System are talking about a swimming pool, but they just say ‘pool’ and its meaning can be inferred from the context. As a result, DS-Span can only extract ‘pool’ as a value, which is not sufficient. In the third example, all the predictions are semantically correct; however, in terms of the string match, only DS-Picklist can correctly predict the value. The two other methods rely on span extraction. This is caused by formatting issues; that is, it is not always guaranteed that strings in the context satisfy desired formats, such as time expressions. Based on our analysis, future work needs to consider more relevant evaluation metrics than the widely-used string matching metric. For example, in the QA research community, it is investigated how to more robustly evaluate QA models (Chen et al., 2019).

Open discussions Multi-domain dialog state tracking is enjoying popularity in enhancing research on task-oriented dialog systems, to handle

User	i am looking for an expensive place to stay on the north side of cambridge .
System	i am sorry , i haven ' t found any matches , would you like me to look for something else ?
User	i am looking for a 4 star hotel and i need free internet and parking .
Ground Truths	<hotel, internet, yes>, <hotel, stars, 4>, <hotel, parking, yes>, <hotel, type, hotel>, <hotel, area, north>, <hotel, price range, expensive>
DS-Span	<hotel, internet, free internet>, <hotel, stars, 4>, <hotel, parking, internet>, <hotel, type, none>, <hotel, area, north>, <hotel, price range, expensive>
DS-DST	<hotel, internet, yes>, <hotel, stars, 4>, <hotel, parking, yes>, <hotel, type, none>, <hotel, area, north>, <hotel, price range, expensive>
User	it's so hot today , can you help me find a good pool to visit on the north side of the city ?
System	i have 2 pools in the north area of town : jesus green outdoor pool and kings hedges learner pool . which do you prefer ?
User	kings hedges sounds nice . can i get the address please ?
Ground Truths	<attraction, area, north>, <attraction, type, swimming pool>, <attraction, name, kings hedges learner pool>
DS-Span	<attraction, area, north>, <attraction, type, pool>, <attraction, name, kings hedges learner pool>
DS-DST	<attraction, area, north>, <attraction, type, swimming pool>, <attraction, name, kings hedges learner pool>
User	do you happen to know of any trains leaving for cambridge this wednesday ?
System	yes . there are a total of 202 trains leaving for cambridge on wednesday . where will you be departing from ?
User	i will be leaving from norwich and i need to arrive by 8 : 15 .
System	the tr4203 is leaving from norwich to cambridge at 05 : 16 . would you like to book a ticket ?
User	sure , can you book that for 2 people and provide my reference number ?
Ground Truths	<train, arrive by, 08 : 15>, <train, departure, norwich>, <train, day, wednesday>, <train, book people, 2>, <train, destination, cambridge>
DS-Span	<train, arrive by, 8 : 15>, <train, departure, norwich>, <train, day, wednesday>, <train, book people, 2>, <train, destination, cambridge>
DS-DST	<train, arrive by, 8 : 15>, <train, departure, norwich>, <train, day, wednesday>, <train, book people, 2>, <train, destination, cambridge>
DS-Picklist	<train, arrive by, 08 : 15>, <train, departure, norwich>, <train, day, wednesday>, <train, book people, 2>, <train, destination, cambridge>

Table 6: Predicted dialog states on the MultiWOZ 2.1 validation set, bold face means incorrect prediction. The first two examples show comparisons between DS-Span and DS-DST. The last example shows comparisons between DS-Span, DS-DST and DS-Picklist.

tasks across different domains and support a large number of services. However, it should be noted that there is much room for improvement with the popular MultiWOZ 2.0 (Budzianowski et al., 2018) and MultiWOZ 2.1 (Eric et al., 2019) datasets, due to their annotation errors, ambiguity, and inconsistency. Moreover, a potential problem is that no standard ways have been established for the evaluation of the MultiWOZ dataset. Some papers are following the pre-processing ways provided by Wu et al. (2019), while others have their own ways, which may result in unfair comparisons; for example, there are some ‘none’ values in the test set, and an evaluation metric without considering them will lead to higher accuracy (up to 3% in our experience). Recent work has refined the datasets to form the latest MultiWOZ 2.2 dataset with higher quality and consistency (Zang et al., 2020). We encourage providing more details of the data processing in future work, and more importantly, testing models on the test set with the higher quality.

6 Conclusion

In this paper, we have proposed a dual strategy model with strong interactions between the dialog context and domain-slot pairs for the task of

multi-domain dialog state tracking. In particular, we predict the slot value via selecting over a partial ontology for categorical slots or finding values from the dialog context for non-categorical slots. Our models achieve the state of the art results on the MultiWOZ 2.0 and competitive results on the MultiWOZ 2.1. Moreover, we conduct a comprehensive analysis on the dataset to facilitate future research.

Acknowledgments

This work is supported in part by NSF under grants III-1763325, III-1909323, and SaTC-1930941. We thank Salesforce research members for their insightful discussions, and the anonymous reviewers for their helpful and valuable comments.

References

- Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering. *arXiv preprint arXiv:1911.10470*.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-

- scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*.
- Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*.
- Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *AAAI*, pages 7521–7528.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.
- Jianfeng Gao, Michel Galley, Lihong Li, et al. 2019a. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298.
- Shuyang Gao, Abhishek Sethi, Sanchit Aggarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019b. Dialog state tracking: A neural reading comprehension approach. In *SIGDIAL*.
- Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014a. The second dialog state tracking challenge. In *SIGDIAL*.
- Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014b. The third dialog state tracking challenge. In *IEEE SLT*.
- Matthew Henderson, Blaise Thomson, and Steve Young. 2014c. Word-based dialog state tracking with recurrent neural networks. In *SIGDIAL*.
- Seokhwan Kim, Luis Fernando D’Haro, Rafael E Banchs, Jason D Williams, Matthew Henderson, and Koichiro Yoshino. 2016. The fifth dialog state tracking challenge. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 511–517. IEEE.
- Seokhwan Kim, Luis Fernando D’Haro, Rafael E Banchs, Jason D Williams, and Matthew Henderson. 2017. The fourth dialog state tracking challenge. In *Dialogues with Social Robots*, pages 435–449. Springer.
- Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, et al. 2019. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*.
- Adarsh Kumar, Peter Ku, Anuj Kumar Goyal, Angeliki Metallinou, and Dilek Hakkani-Tur. 2020. Ma-dst: Multi-attention based scalable dialog state tracking. *AAAI*.
- Hung Le, Richard Socher, and Steven CH Hoi. 2020. Non-autoregressive dialog state tracking. *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 *ACL*.
- Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. In *NIPS*.
- Bing Liu and Ian Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. In *INTERSPEECH*.
- Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience.
- Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *ACL*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf*.
- Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *ACL*.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *AAAI*.
- Liliang Ren, Jianmo Ni, and Julian McAuley. 2019. Scalable and accurate dialogue state tracking via hierarchical sequence generation. *EMNLP*.
- Yong Shan, Zekang Li, Jinchao Zhang, Fandong Meng, Yang Feng, Cheng Niu, and Jie Zhou. 2020. A contextual hierarchical attention network with adaptive objective for dialogue state tracking. *arXiv preprint arXiv:2006.01554*.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*.

- Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *SIGDIAL*.
- Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogues. *arXiv preprint arXiv:2004.06871*.
- 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*.
- Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.
- Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. *arXiv preprint arXiv:2007.12720*.
- Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *ACL*.
- Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *arXiv preprint arXiv:1911.06192*.

A Appendix

A.1 Training Details

We employ a pre-trained BERT model with the “bert-base-uncased” configuration.⁴ During the fine-tuning process, we update all the model parameters using the BertAdam (Devlin et al., 2019) optimizer. The maximum input sequence length after WordPiece tokenization for BERT is set to 512. The constant margin λ is set to 0.5 for the DS-DST and DS-Picklist models. For the MultiWOZ 2.0 and MultiWOZ 2.1 datasets, the initial learning rate is set to $1e - 4$, and the proportion for learning rate warmup is set to 0.1. The batch size is set to 16. We follow the learning rate decay mechanism as in (Lee et al., 2019), and we set the maximum number of the training epochs to 5. We check the joint accuracy on the validation set after every 1,000 iterations, and keep the best checkpoint to be used for the final evaluation on the test set.

A.2 Sample Output

We present outputs of DS-Span and DS-DST in all the turns for two dialogues (i.e., *MUL0729*, *PMUL2428*) on the validation set of the MultiWOZ 2.1. Table 7 and Table 8 show the predicted dialog states for *MUL0729* and *PMUL2428*, respectively. In Table 7, *hotel type* and *hotel internet* are predicted incorrectly by DS-Span, where the value *yes* of *hotel internet* has a different description *free wifi* in the dialog context. For this type of values, DS-Span cannot find the spans directly in the dialog context. In Table 8, DS-Span does not correctly predict the state *<taxi, departure, funky fun house>* at the 6_{th} turn.

⁴<https://github.com/huggingface/transformers/tree/master/examples>

Turn 1	User:	i am looking for a train from ely to cambridge . do you have such trains ?
	Ground Truths:	<train, destination, cambridge>, <train, departure, ely>
	DS-Span:	<train, destination, cambridge>, <train, departure, ely>
	DS-DST:	<train, destination, cambridge>, <train, departure, ely>
Turn 2	System:	there are 70 trains do you have a date and time you would like to go ?
	User:	i would like to leave on a sunday and arrive by 11:30 .
	Ground Truths:	<train, arrive by, 11:30>, <train, destination, cambridge>, <train, departure, ely>, <train, day, sunday>
	DS-Span:	<train, arrive by, 11:30>, <train, destination, cambridge>, <train, departure, ely>, <train, day, sunday>
Turn 3	DS-DST:	<train, arrive by, 11:30>, <train, destination, cambridge>, <train, departure, ely>, <train, day, sunday>
	System:	tr4212 is arriving at 9:52 would that work for you ?
	User:	what time does that train depart ?
	Ground Truths:	<train, arrive by, 11:30>, <train, destination, cambridge>, <train, departure, ely>, <train, day, sunday>
Turn 4	DS-Span:	<train, arrive by, 11:30>, <train, destination, cambridge>, <train, departure, ely>, <train, day, sunday>
	DS-DST:	<train, arrive by, 11:30>, <train, destination, cambridge>, <train, departure, ely>, <train, day, sunday>
	System:	the train leaves at 9:35 . shall i book it for you ?
	User:	no , thank you . i would like to see if you could find a place for me to stay that would have a 4 star rating .
Turn 5	Ground Truths:	<train, day, sunday>, <hotel, stars, 4>, <train, destination, cambridge>, <train, arrive by, 11:30>, <train, departure, ely>
	DS-Span:	<train, day, sunday>, <hotel, stars, 4>, <train, destination, cambridge>, <train, arrive by, 11:30>, <train, departure, ely>
	DS-DST:	<train, day, sunday>, <hotel, stars, 4>, <train, destination, cambridge>, <train, arrive by, 11:30>, <train, departure, ely>
	System:	of course , there are many 4 star hotel -s to choose from . would you like me to book 1 for you ?
Turn 6	User:	not yet , thanks . are any of them guesthouses ?
	Ground Truths:	<train, day, sunday>, <hotel, stars, 4>, <train, destination, cambridge>, <train, arrive by, 11:30>, <train, departure, ely>, <hotel, type, guest house>
	DS-Span:	<train, day, sunday>, <hotel, stars, 4>, <train, destination, cambridge>, <train, arrive by, 11:30>, <train, departure, ely>, <u><hotel, type, hotel></u>
	DS-DST:	<train, day, sunday>, <hotel, stars, 4>, <train, destination, cambridge>, <train, arrive by, 11:30>, <train, departure, ely>, <hotel, type, guest house>
Turn 7	System:	there are 18 guesthouses to choose from , do you have a preference to the area you would like to stay ?
	User:	i need a 4 star , and in the east with free wifi for 4 people , 5 nights . i'll need a reference number .
	Ground Truths:	<train, day, sunday>, <hotel, book stay, 5>, <hotel, book people, 4>, <hotel, stars, 4>, <train, destination, cambridge>, <hotel, internet, yes>, <train, arrive by, 11:30>, <train, departure, ely>, <hotel, area, east>, <hotel, type, guest house>
	DS-Span:	<train, day, sunday>, <hotel, book stay, 5>, <hotel, book people, 4>, <hotel, stars, 4>, <train, destination, cambridge>, <u><hotel, internet, no></u> , <train, arrive by, 11:30>, <train, departure, ely>, <hotel, area, east>, <u><hotel, type, hotel></u>
Turn 8	DS-DST:	<train, day, sunday>, <hotel, book stay, 5>, <hotel, book people, 4>, <hotel, stars, 4>, <train, destination, cambridge>, <hotel, internet, yes>, <train, arrive by, 11:30>, <train, departure, ely>, <hotel, area, east>, <hotel, type, guest house>
	System:	do you want that guesthouse reservation to begin on sunday ?
	User:	yes . i need 5 nights starting on sunday .
	Ground Truths:	<train, day, sunday>, <hotel, book stay, 5>, <hotel, book people, 4>, <hotel, stars, 4>, <train, destination, cambridge>, <hotel, internet, yes>, <hotel, type, guest house>, <train, arrive by, 11:30>, <train, departure, ely>, <hotel, area, east>, <hotel, book day, sunday>
Turn 9	DS-Span:	<train, day, sunday>, <hotel, book stay, 5>, <hotel, book people, 4>, <hotel, stars, 4>, <train, destination, cambridge>, <hotel, internet, yes>, <hotel, type, hotel>, <train, arrive by, 11:30>, <train, departure, ely>, <hotel, area, east>, <hotel, book day, sunday>
	DS-DST:	<train, day, sunday>, <hotel, book stay, 5>, <hotel, book people, 4>, <hotel, stars, 4>, <train, destination, cambridge>, <hotel, internet, yes>, <hotel, type, guest house>, <train, arrive by, 11:30>, <train, departure, ely>, <hotel, area, east>, <hotel, book day, sunday>
	System:	i have confirmed your reservation at allenbell starting on sunday for 5 nights . your reference number is sltivabu . can i help with anything else ?
	User:	that is all i need . thank you so much for all your help .
Turn 10	Ground Truths:	<train, day, sunday>, <hotel, book stay, 5>, <hotel, book people, 4>, <hotel, stars, 4>, <train, destination, cambridge>, <hotel, internet, yes>, <hotel, type, guest house>, <train, arrive by, 11:30>, <train, departure, ely>, <hotel, area, east>, <hotel, book day, sunday>
	DS-Span:	<train, day, sunday>, <hotel, book stay, 5>, <hotel, book people, 4>, <hotel, stars, 4>, <train, destination, cambridge>, <hotel, internet, yes>, <u><hotel, type, hotel></u> , <train, arrive by, 11:30>, <train, departure, ely>, <hotel, area, east>, <hotel, book day, sunday>
	DS-DST:	<train, day, sunday>, <hotel, book stay, 5>, <hotel, book people, 4>, <hotel, stars, 4>, <train, destination, cambridge>, <hotel, internet, yes>, <hotel, type, guest house>, <train, arrive by, 11:30>, <train, departure, ely>, <hotel, area, east>, <hotel, book day, sunday>
	System:	

Table 7: Predicted dialog states of DS-Span and DS-DST for domains (i.e., *train*, *hotel*) in dialogue *MUL0729* from the MultiWOZ 2.1 dataset.

Turn 1	User:	i am planning a trip to go to a particular restaurant , can you assist ?
	Ground Truths:	
	DS-Span:	
	DS-DST:	
Turn 2	System:	sure , what is the name of this particular restaurant ?
	User:	it s called nandos and i would like to book it for monday at 15:00 . there will be 6 people .
	Ground Truths:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book time, 15:00>, <restaurant, book people, 6>
	DS-Span:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book time, 15:00>, <restaurant, book people, 6>
Turn 3	DS-DST:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book time, 15:00>, <restaurant, book people, 6>
	System:	no problem ! i have your table reserved for 15:00 on monday . they will hold your table for 15 minutes , your reference number is hvb51vam .
	User:	thank you . am also looking for place -s to go in town . the attraction should be in the east
	Ground Truths:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>
Turn 4	DS-Span:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>
	DS-DST:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>
	System:	what are you looking to do ? my system has located 10 place -s .
	User:	could you recommend something ?
Turn 5	Ground Truths:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>
	DS-Span:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>
	DS-DST:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>
	System:	yes , the funky fun house is a great place .
Turn 6	User:	great ! can you tell me what the entrance fee is please ?
	Ground Truths:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>, <attraction, name, funky fun house>
	DS-Span:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>, <attraction, name, funky fun house>
	DS-DST:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>, <attraction, name, funky fun house>
Turn 7	System:	no , i am sorry . you will have to call them for the entrance fee . the phone number is 01223304705 .
	User:	thank you . i am also looking to book a taxi to travel between the 2 . i need it to arrive to the restaurant by the reservation time .
	Ground Truths:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>, <taxi, arrive by, 15:00>, <attraction, name, funky fun house>, <taxi, destination, nandos>, <taxi, departure, funky fun house>
	DS-Span:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>, <taxi, arrive by, 15:00>, <attraction, name, funky fun house>, <taxi, destination, nandos>, <taxi, departure, funky fun house>
Turn 8	DS-DST:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>, <taxi, arrive by, 15:00>, <attraction, name, funky fun house>, <taxi, destination, nandos>, <taxi, departure, funky fun house>
	System:	your taxi is booked and will be a white audi . the contact number is 07057575130 . how else may i help you ?
	User:	that s all . thank you for your help !
	Ground Truths:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>, <taxi, arrive by, 15:00>, <attraction, name, funky fun house>, <taxi, destination, nandos>, <taxi, departure, funky fun house>
Turn 9	DS-Span:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>, <taxi, arrive by, 15:00>, <attraction, name, funky fun house>, <taxi, destination, nandos>, <taxi, departure, funky fun house>
	DS-DST:	<restaurant, book day, monday>, <restaurant, name, nandos>, <restaurant, book people, 6>, <attraction, area, east>, <restaurant, book time, 15:00>, <taxi, arrive by, 15:00>, <attraction, name, funky fun house>, <taxi, destination, nandos>, <taxi, departure, funky fun house>
	System:	
	User:	

Table 8: Predicted dialog states of DS-Span and DS-DST for domains (i.e., *taxi*, *attraction*, *restaurant*) in dialogue *PMUL2428* from the MultiWOZ 2.1 dataset.