# END-TO-END MULTI-DOMAIN TASK-ORIENTED DIALOGUE SYSTEMS WITH MULTI-LEVEL NEURAL BELIEF TRACKER

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

It has been an open research challenge for developing an end-to-end multi-domain task-oriented dialogue system, in which a human can converse with the dialogue agent to complete tasks in more than one domain. First, tracking belief states of multi-domain dialogues is difficult as the dialogue agent must obtain the complete belief states from all relevant domains, each of which can have shared slots common among domains as well as unique slots specifically for the domain only. Second, the dialogue agent must also process various types of information, including contextual information from dialogue context, decoded dialogue states of current dialogue turn, and queried results from a knowledge base, to semantically shape context-aware and task-specific responses to human. To address these challenges, we propose an end-to-end neural architecture for task-oriented dialogues in multiple domains. We propose a novel Multi-level Neural Belief Tracker which tracks the dialogue belief states by learning signals at both slot and domain level independently. The representations are combined in a Late Fusion approach to form joint feature vectors of *(domain, slot)* pairs. Following recent work in end-to-end dialogue systems, we incorporate the belief tracker with generation components to address end-to-end dialogue tasks. We achieve state-of-the-art performance on the MultiWOZ2.1 benchmark with 50.91% joint goal accuracy and competitive measures in task-completion and response generation.

## 1 INTRODUCTION

In a task-oriented dialogue system, the Dialogue State Tracking (DST) module is responsible for updating dialogue states (essentially, what the user wants) at each dialogue turn. The DST supports the dialogue agent to steer the conversation towards task completion. As defined by Henderson et al. (2014a), a dialogue belief state consists of *inform* slots - information to query a given knowledge base or database (DB), and *request* slots - information to be returned to the users. Task-oriented dialogues can be categorized as either single-domain or multi-domain dialogues. In single-domain dialogues, humans converse with the dialogue agent to complete tasks of one domain. In contrast, in multi-domain dialogues, the tasks of interest can come from different domains. A dialogue state in a multi-domain dialogue should include all *inform* and *request* slots of corresponding domains up to the current turn. Examples of a single-domain dialogue and a multi-domain dialogue with annotated states after each turn can be seen in Figure 1.

Despite there being several efforts in developing task-oriented dialogue systems in a single domain (Wen et al., 2016a; 2017; Li et al., 2017; Lei et al., 2018), there have been limited contributions for multi-domain task-oriented dialogues. Developing end-to-end systems for multi-domain dialogues faces several challenges: (1) Belief states in multi-domain dialogues are usually larger and more complex than in single-domain, because of the diverse information from multiple domains. Each domain can have shared slots that are common among domains or unique slots that are not shared with any. (2) In an end-to-end system, the dialogue agent must incorporate information from source sequences, e.g. dialogue context and human utterances, as well as tracked belief states and extracted information from knowledge base, to semantically shape a relevant response with accurate information for task completion.

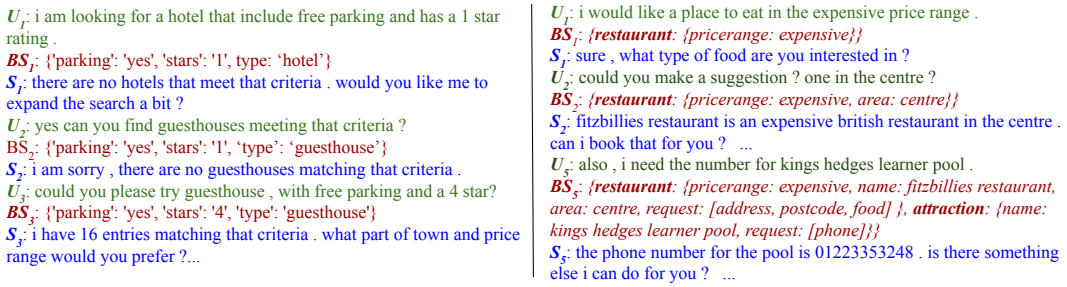


Figure 1: Examples of single-domain (Left) and multi-domain dialogues (Right). $U$: user utterance, $BS$: dialogue belief state, $S$: system response. The subscript indicates dialogue step. Best viewed in color.

Directly applying methods for single-domain dialogues to multi-domain dialogues is not straightforward because the belief states extend across multiple domains. A possible solution is to process a multi-domain dialogue for $N_D$ times for $N_D$ domains, each time obtaining a belief state of one domain. However, this approach does not allow learning co-references in dialogues whereby users can switch from one domain to another turn by turn. We propose an end-to-end dialogue system approach which explicitly track the dialogue states in multiple domains altogether. Specifically, (1) we propose Multi-level Neural Belief Tracker to process contextual information for both slot-level and domain-level signals independently. The two levels are subsequently combined to learn multi-domain dialogue states. Our dialogue state tracker enables shared learning of slots common among domains as well as learning of unique slots in each domain. (2) we utilize multi-head attention layers (Vaswani et al., 2017) to comprehensively process various types of information: dialogue context, user utterances, belief states of both *inform* and *request* slots, and DB query results. The multi-head structure allows the model to independently attend to the features over multiple representation sub-spaces; and (3) we combine all components to create a dialogue system from state tracking to response generation. The system can be jointly learned in an end-to-end manner. Our end-to-end dialogue system utilizes supervision signals of dialogue states and output responses without using system action annotation.

To comprehensively validate our method, we compare our models with baselines in end-to-end, DST, and context-to-text generation settings. We achieve the state-of-the-art performance in DST, task-completion, and response generation in the MultiWOZ2.1 corpus (Budzianowski et al., 2018; Eric et al., 2019) as compared to other baselines in similar settings. In context-to-text generation setting that allows supervision of dialogue acts, our models can achieve competitive measures of Inform and BLEU metric.

## 2 Related Work

Our work is related to 2 main bodies of research: DST and end-to-end dialogue systems.

### 2.1 Dialogue State Tracking

Prior DST work focuses on single-domain dialogues using WOZ (Wen et al., 2017) and DSTC2 (Henderson et al., 2014a) corpus. (Mrkšić et al., 2015; Wen et al., 2016b; Rastogi et al., 2017) address transfer learning in dialogues from one domain to another rather than multiple domains in a single dialogue. Our work is more related to recent effort for multi-domain DST such as (Ramadan et al., 2018; Lee et al., 2019; Wu et al., 2019a; Goel et al., 2019). These models can be categorized into two main categories of DST: fixed-vocabulary and open-vocabulary approach. Fixed vocabulary models (Zhong et al., 2018; Ramadan et al., 2018; Lee et al., 2019) assume known slot ontology with a fixed candidate set for each slot. Open-vocabulary models (Lei et al., 2018; Wu et al., 2019a; Gao et al., 2019) derive the candidate set based on the source sequence i.e. dialogue history, itself. Our approach is more related to open-vocabulary approach as we aim to dynamically generate dialogue state based on input dialogue history. Different from most prior work, our Multi-level Neural Belief Tracker can learn domain-level and slot-level signals independently and both are combined in a Late Fusion manner to obtain contextual representations of all *(domain, slot)* pairs.

### 2.2 END-TO-END DIALOGUE SYSTEMS

Conventionally, an end-to-end dialogue system is composed of separate modules for Natural Language Understanding (NLU) (Hashemi et al., 2016; Gupta et al., 2018), DST (Henderson et al., 2014b; Zhong et al., 2018), Dialogue Policy (Peng et al., 2017; 2018), and Natural Language Generator (NLG) (Wen et al., 2016a; Su et al., 2018). These components can be learned independently and combined into a pipeline architecture for end-to-end system (Wen et al., 2017; Liu & Lane, 2017; Li et al., 2017). Another line of research aims to develop a dialogue agent without modularizing these components but incorporating them into a single network (Eric & Manning, 2017; Lei et al., 2018; Madotto et al., 2018; Wu et al., 2019b). Our work is more related to the latter approach whereby we incorporate conventional components into an integrate network architecture and jointly train all parameters. However, following (Lei et al., 2018), we consider a separate module that combines NLU and DST together. The module utilizes additional supervision for more fine-grained tracking of user goals. This strategy is also suitable for large-scale knowledge base with large number of entities. (Madotto et al., 2018; Wu et al., 2019b; Gangi Reddy et al., 2019) completely omit the DST component by formulating entity attributes into memory form based on *(Subject, Relation, Object)* tuples. These models achieve good performance in small-scale corpus such as In-Car Assistant (Eric & Manning, 2017) and WOZ2.0 (Wen et al., 2017) but will become extremely hard to scale to large knowledge base in multi-domain setting such as MultiWOZ corpus.

## 3 APPROACH

Given a dialogue with dialogue history of $t-1$ turns, each including a pair of user utterance and system response, $(U_1, S_1), ..., (U_{t-1}, S_{t-1})$, the user utterance at current dialogue turn $U_t$, and a knowledge base in form of entity data tables, the goal of a task-oriented dialogue system is to generate a response $S_t$ that is not only appropriate to the dialogue context, but also task-related with the correct entity results for the users. In the multi-domain dialogue setting, turns in the dialogue history and the current user utterance could come from different domains. Therefore, the generated response in this setting should also be domain-related with the correct domain-specific information for the users. We propose a novel Multi-level Neural Belief Tracker to track belief states at both domain level and slot level to address multi-domain dialogues. Following (Lei et al., 2018), we utilize the previous belief states $B_{t-1}$ as an input to the model. This allows the model to rely on the dialogue states detected from the previous dialogue step $t-1$ to update the state of the current step $t$. In addition, we adopt the attention-based principle of Transformer network (Vaswani et al., 2017) and propose an end-to-end architecture for task-oriented dialogues. Our model allows comprehensive information processing from different input sources, incorporating contextual features from dialogue context and user utterance as well as learning signals from domain-level and slot-level dialogue states.

Our solution consists of 3 major components: (i) *Encoders* encode sequences of dialogue history, current user utterances, target system responses, domain and slot names, and previous dialogue belief states, into continuous representations. (ii) *Multi-level Neural Belief Tracker* includes 2 modules, one for processing slot-level information and one for domain-level information. Each module comprises attention layers to project domain or slot token representations and attend on relevant features for state tracking. The outputs of the two modules are combined to create domain-slot joint feature representations. Each feature representation is used as a context-aware vector to decode the corresponding *inform* or *request* slots in each domain. (iii) *Response Generator* projects the target system responses and incorporates contextual information from dialogue context as well as intermediate variables from the state tracker and DB query results. Employing attention mechanisms with feed-forward and residual connections allows our models to focus on relevant parts of the inputs and pass on the relevant information to decode appropriate system responses. We combine all the modules into an end-to-end architecture and jointly train all components. An overview of the proposed approach can be seen in Figure 2.

### 3.1 ENCODERS

An encoder encodes a text sequence of tokens $(x_1, ..., x_n)$ to a sequence of continuous representation $z = (z_1, ..., z_n) \in \mathbb{R}^{n \times d}$. Each encoder includes a token-level trainable embedding layer and layer normalization (Ba et al., 2016). Depending on the type of text sequences, we inject sequential characteristics of the tokens (i.e. their positions in the sequence) using a sine and cosine positional encoding functions (Vaswani et al., 2017). Element-wise summation is used to combine the token-

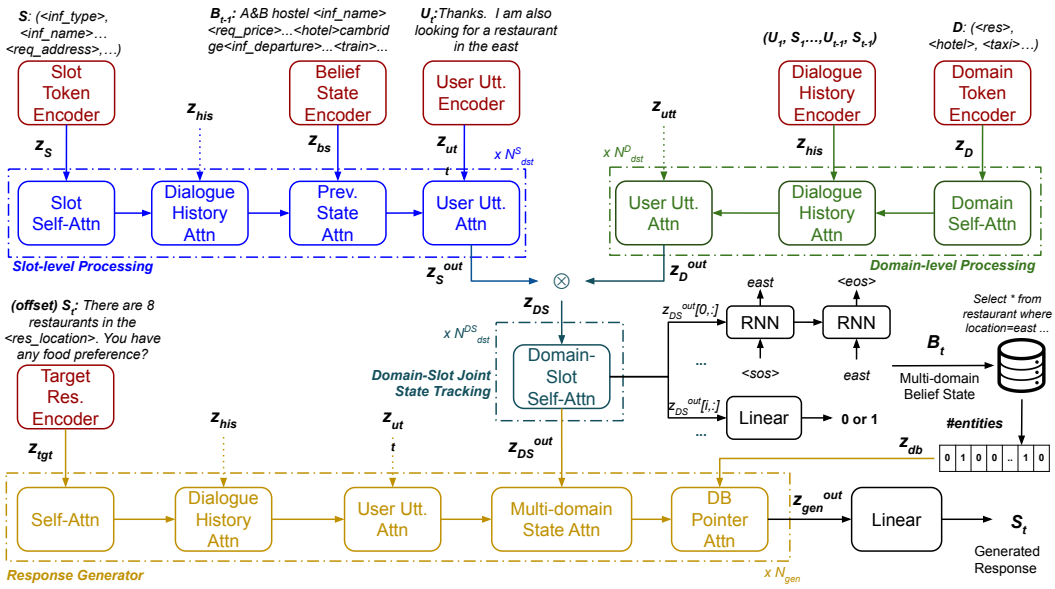


Figure 2: Our architecture consists of 3 major components: (i) *Encoders* encode sequences of dialogue history, previous user utterances, domain and slot tokens, and dialogue belief states of previous turn, into continuous representations; (ii) *Multi-level Neural Belief Tracker* consists of 2 modules, one for learning slot-level signals and one for domain-level signals; the outputs are combined to obtain domain-slot joint features, which are used decode multi-domain belief states; and (iii) *Response Generator* incorporates information from dialogue context, dialogue states, and knowledge base, to decode system responses. For simplicity, Feed Forward, Residual Connection, and Layer Normalization layers are not presented. Best viewed in color.

level embedding with positional embedding, each has the same embedding dimension $d$. The current user utterance $U_t$ is tokenized, prepended and appended with $\langle sos \rangle$ and $\langle eos \rangle$ token respectively. In the dialogue history, each human utterance and system response up to dialogue step $t-1$ is processed similarly. The tokenized past utterances and system responses are concatenated sequentially by the dialogue step. For target system response $S_t$, during training, the sequence is offset by one position to ensure that token prediction in generation step $i$ is based on the previous positions only i.e. $1, ..., i-1$.

Denoting $name_{slot_i}$ and $value_{slot_i}$ as the slot name and slot value of slot $i$, we create sequences of dialogue belief state from previous turn by following the template: $value_{slot_i}$ $\langle inf\_name_{slot_i} \rangle ... \langle req\_name_{slot_j} \rangle ... \langle domain_d \rangle ...$ A $\langle req\_name_{slot_j} \rangle$ is only included in the sequence if $slot_j$ is predicted as in the previous turn. As a slot such as *area* can be both *request* or *inform* type, the 2 slot types are differentiated by the prefixes $inf$ and $req$. Our belief sequences can be used to encode past dialogue states of multiple domains, each separated by the $\langle domain_d \rangle$ token. To learn slot-level and domain-level signals for state tracking, we construct set of slot and domain tokens as input to the state tracker. Each input set is created by concatenating slot names or domains: $S = (...\langle inf\_name_{slot_i} \rangle, ..., \langle req\_name_{slot_j} \rangle ...), \|S\| = N_S^{inf} + N_S^{req} = N_S$ , and $D = (\langle domain_1 \rangle, ..., \langle domain_{N_D} \rangle)$ respectively. Both sequences are kept fixed in all dialogue samples to factor in all possible domains and slots for multi-domain state tracking. Positional encoding is used in all sequences except for input sets of slot and domain tokens as these sets do not contain sequential characteristic. Embedding weights are shared among all the encoders of source sequences. Embedding weights of the target system responses are not shared to allow the models to learn the semantics of input and output sequences differently.

### 3.2 Multi-level Neural Belief Tracker

The DST module processes slot-level and domain-level information independently, and integrates the two for multi-domain state tracking. We adopt a Late Fusion approach to combine domain and slot representations in deeper network layers.

**Slot-level Processing**. Given the encoded features from the source sequences, including dialogue history $z_{his}$, previous belief state $z_{bs}$, and the current user utterance $z_{utt}$, the slot-level signals are learned by projecting the encoded slot token sequence $z_S$ through $N_{dst}^S$ identical layers. Each layer contains 4 attention blocks, each of which employ the multi-head attention mechanism (Vaswani

et al., 2017) to attend on the inputs at different representation sub-spaces. Each attention block is coupled with a position-wise feed-forward layer, including 2 linear transformations with ReLU activation in between. Residual connection (He et al., 2016) and layer normalization (Ba et al., 2016) are employed in each attention block. Specifically, given the current feature vector $z_S^{out}$ as output from previous attention block (or $z_S$ itself in the first attention block of the first processing layer) and the encoded features $z_{seq}$ of a source sequence, the multi-head attention is defined as:

$$m = \text{Concat}(h_1, ..., h_{N_h})W^O \tag{1}$$

$$h_i = \text{Attn}(z_S^{out}W_i^Q, z_{seq}W_i^K, z_{seq}W_i^V) \text{ where } \text{Attn}(q, k, v) = \text{softmax}(\frac{qk^T}{\sqrt{d_a}})v \tag{2}$$

where $W_i^Q\, W_i^K, W_i^V \in \mathbb{R}^{d \times d_a}, W^O \in \mathbb{R}^{N_h d_a \times d}$, and $seq = \{S, his, bs, utt\}$ (for simplicity, the subscripts of $S$ and $seq$ are omitted in each $W$). The first attention block is a self-attention, i.e. $seq = S$, which enables learning the relation between slots independently from domains. Subsequent attention layers on dialogue context, previous belief state, and user utterance of current turn, inject each slot token representation with dialogue contextual information up to current user utterance in turn $t$. Through residual connection, the contextual information are passed forward in each $z_S^{out}$. Using different attention blocks allows flexible processing of information from various input sources.

**Domain-level Processing**. The input to the domain-level processing module includes the encoded domain token sequence $z_D$, the encoded dialogue history up to turn $t-1$ $z_{his}$, and the encoded user utterance of current turn $z_{utt}$. The domain features are passed through $N_{dst}^D$ identical layers, each of which include 3 multi-head attention blocks to obtain important contextual information from dialogue context and user utterance. Similarly to slot-level processing, a self-attention block is leveraged to allow reasoning among domains independently from slots. Attending on dialogue history and current user utterance separately enables learning domain signals from the contextual information of past dialogue turns and current turns differently. Therefore, the models can potentially detect changes of dialogue domains from past turns to the current turn. Especially in multi-domain dialogues, users can switch from one domain to another and the generated responses should address the latest domain.

**Domain-Slot Joint State Tracking**. The output slot token representations $z_S^{out} \in \mathbb{R}^{N_S \times d}$ and domain token representations $z_D^{out} \in \mathbb{R}^{N_D \times d}$ are combined by expanding the representations to identical dimensions and using element-wise multiplication, resulting in domain-slot joint features $z_{DS} \in \mathbb{R}^{N_S N_D \times d}$. The joint features are passed through $N_{dst}^{DS}$ identical self-attention layers to allow learning over all applicable $N_D \times N_S$ combinations of domains and slots. In each attention block, we mask the joint features at positions of inapplicable domain-specific slots. For example, there is no *inform_departure* slot in the *hotel* domain and its position is masked in each $z_{DS}^{out}$. The output vectors at the last layer are used as context-aware and domain-specific representations to decode dialogue states. Each feature vector $z_{DS}^{out}[i, :] \in \mathbb{R}^d$ is used to decode the corresponding domain-specific slot $i$. The vector is used as initial hidden state for an RNN decoder to decode an *inform* slot token by token or passed through a linear transformation layer for binary classification for a *request* slot.

### 3.3 Response Generator

The decoded dialogue states are used to query the DBs of all domains and obtain the number of the result entities in each domain. We then create a fixed-dimensional one-hot pointer vector for each domain $d$: $z_{db}^d \in \{0, 1\}^6$ and $\sum_i^6 z_{db,i}^d = 1$. Each position of the pointer vector indicates the number of result entities. The pointer vectors of all domains are concatenated to create a multi-domain pointer vector $z_{db} \in \mathbb{R}^{6N_D}$. We embed the pointer vector with the learned embedding and positional embedding as similarly described in Section 3.1, resulting in $z_{db} \in \mathbb{R}^{6N_D \times d}$. The DB pointer vector $z_{db}$, context-aware domain-slot joint features $z_{DS}^{out}$, encoded dialogue history $z_{his}$, and user utterance of current turn $z_{utt}$, are used as inputs to incorporate relevant signals to decode system responses. The generator includes $N_{gen}$ identical layers, each includes 5 multi-head attention blocks, including a self-attention block at the beginning. Adopting attention with residual connection in each block allows the models to comprehensively obtain contextual cues, either through text sequences or domain-slot joint features, and knowledge base signals from DB pointer vectors. The final output $z_{gen}^{out}$ is passed to a linear transformation with softmax activation to decode system responses.

### 3.4 END-TO-END TRAINING

The objective function is a combination of belief state objectives, including the log-likelihood of all *inform* slot sequences $S^{inf}$, and the binary cross entropy of *request* slots $S^{req}$, and the system response objective, including the log-likelihood of the target response sequence $T$, as follows:

$$L = L(T) + L(B_t) = L(T) + L(B_t^{inf}) + L(B_t^{req}) = \sum_m \log P(y_m|y_{m-1}, ..., y_1, C, U, B_t, Q) + \sum_i \sum_n \log P(s_n^i|s_{n-1}^i, ..., s_1^i, C, U, B_{t-1}) + \sum_r \log P(s^r|C, U, B_{t-1}) \quad (3)$$

where $i = [1, N_S^{inf}]$ and $r = [1, N_S^{req}]$. The above objectives are conditioned on the input features, including dialogue context $C$, current user utterance $U$, previous and current belief state $B_{t-1}$ and $B_t$, and DB queries $Q$.

## 4 EXPERIMENTS

### 4.1 DATA

We used the MultiWOZ 2.1 dataset (Budzianowski et al., 2018; Eric et al., 2019) which consists of both single-domain and multi-domain dialogues. Compared to version 2.0, MultiWOZ 2.1 is improved with some correction of DST labels, including about 40% changes across training samples. We pre-processed the dialogues by tokenizing, lower-casing, and delexicalizing all system responses. From the belief state annotation of the training data, we identified all possible domains and slots. We identified $N_D = 7$ domains and $N_S = 35$ unique *inform* slots in total. We followed the pre-processing scripts as provided by (Budzianowski et al., 2018; Wu et al., 2019b). The result corpus includes 8,438 dialogues in the training with an average of 1.8 domains per dialogue. Each dialogue has more than 13 turns. There are 1,000 in each validation and test set, each including an average of 1.9 domains per dialogue. Other details of data pre-processing procedures, domains, slots, and entity DBs, are included in Appendix A.1.

### 4.2 TRAINING SETUP

The model parameters are: $d = 256$, $d_a = 32$, $N_h = 8$, $N_{dst}^S = N_{dst}^D = 4$, $N_{dst}^{DS} = 1$, $N_{gen} = 4$. We employed dropout (Srivastava et al., 2014) of $0.3$ at all network layers except the linear layers in the generative components. Label smoothing (Szegedy et al., 2016) for target system responses is applied during training. During training, we utilize teacher-forcing learning strategy by simply using the ground-truth inputs of dialogue state from previous turn and the gold DB pointer. During inference, in each dialogue, we decode system responses sequentially turn by turn, using the previously decoded belief state as input in the current turn, and at each turn, using the decoded belief state to query DBs for pointer vectors. We train all networks in an end-to-end manner with Adam optimizer (Kingma & Ba, 2015) and the learning rate schedule similarly adopted by Vaswani et al. (2017). We used batch size 32 and tuned the *warmup_steps* from 9K to 15K training steps. All models are trained up to $30$ epochs and best models are selected based on validation loss. We used a greedy approach to decode all slots and beam search with beam size $5$ and a length penalty $1.0$ to decode responses. The maximum length is set to 10 tokens for each slot and 20 for system responses. Our models are implemented using PyTorch (Paszke et al., 2017).

### 4.3 RESULTS

To evaluate the models, we use the following metrics: (1) DST metrics: Joint Accuracy and Slot Accuracy (Henderson et al., 2014b). Joint Accuracy compares the predicted dialogue states to the ground truth in each dialogue turn. All slot values must match the ground truth labels to be counted as a correct prediction. Slot Accuracy considers individual slot-level accuracy across the topology. (2) Task-completion metrics: Inform and Success (Wen et al., 2017). Inform refers to system ability to provide an appropriate entity while Success is the system ability to answer all requested attributes. (3) Generation metrics: BLEU score (Papineni et al., 2002). We ran all experiments 3 times and reported the average results. We report results in 2 different settings: end-to-end dialogues and DST. In end-to-end setting, we train a dialogue agent that is responsible for both DST and text generation without assuming access to ground-truth labels.

**End-to-End**. In this setting, we compare our model performance on the joint task of DST and context-to-text generation. For fair comparision, we select TSCP (Lei et al., 2018) as the baseline as TSCP does not use additional supervision signals of system action as input. This is the current state-of-the-art for end-to-end dialogue task in the single-domain WOZ (Wen et al., 2017). TSCP applies pointer network to develop a two-stage decoding process to decode belief states, in a form of text sequence, and subsequently decode system responses. We adapt the method to the multi-domain dialogue setting. We experiment with 2 cases of TSCP in which the maximum length of the output state sequence $L_{bspan}$ is set to 8 and 20 tokens. As can be seen in Table 1, our model outperforms in all metrics, except for the Slot Acc metric in one case. Overall, our model performs well in both multi-domain and single-domain dialogues, especially with higher performance gain in multi-domain dialogues. The performance gain in multi-domain dialogues can be explained by the separate network structure between domain and slot processing modules in our models. This allows our models to learn domain-dependent and slot-dependent signals separately before the two are fused into a joint feature vectors for downstream process. For TSCP, increasing the $L_{bspan}$ from 8 to 20 tokens helps to improve the performance, but also increases the training time to convergence significantly. In our approach, all *inform* and *request* slots are decoded independently and the training time is less affected by the size of the target dialogue states, especially in cases of extensive belief states (e.g. 4 or 5 domains in a dialogue). Additional results by individual domains are described in Appendix A.3.

Table 1: Evaluated on MultiWOZ2.1, the proposed approach achieves better performance than TSCP (Lei et al., 2018), especially with higher performance gain in multi-domain dialogues. The best result in each metric is highlighted in bold.

| Model | Joint Acc | Slot Acc | Inform | Success | BLEU |
|---|---|---|---|---|---|
| *All Dialogues* | | | | | |
| TSCP (L=8) | 31.64% | 95.53% | 45.31% | 38.12% | 11.63 |
| TSCP (L=20) | 37.53% | 96.23% | 66.41% | 45.32% | 15.54 |
| **Ours** | **50.91%** | **97.34%** | **72.45%** | **52.14%** | **19.80** |
| *Multi-domain Dialogues (771 dialogues)* | | | | | |
| TSCP (L=8) | 33.63% | 93.24% | 48.23% | 28.42% | 10.32 |
| TSCP (L=20) | 43.23% | 95.32% | 57.23% | 43.25% | 15.62 |
| **Ours** | **49.53%** | **97.20%** | **66.28%** | **48.19%** | **19.95** |
| *Single-domain Dialogues (229 dialogues)* | | | | | |
| TSCP (L=8) | 53.52% | 98.04% | 75.23% | 46.23% | 12.42 |
| TSCP (L=20) | 55.23% | **98.34%** | 87.23% | 60.23% | 15.02 |
| **Ours** | **58.98%** | 98.18% | **93.36%** | **65.49%** | **18.86** |

**DST**. We isolate the DST components (i.e. training models only with $L(B_t)$) and report the DST performance. We compare the performance with the baseline models on the MultiWOZ 2.1 in Table 2 (Refer to Appendix A.2 for more description of DST baselines). Our model outperforms existing baselines and achieves the state-of-the-art performance in MultiWOZ2.1 corpus. By leveraging on dialogue context signals through independent attention modules at domain level and slot level, our DST can generate slot values more accurately. DST approaches that try to separate domain and slot signals such as TRADE (Wu et al., 2019a) reveal competitive performance. However, our approach has better performance as we enable deeper interaction of context-related signals in each domain and slot representation. Compared to TRADE, our approach can be considered as *Late Fusion* approach that combines representations in deeper network layers for better joint features of domains and slots. We also noted that DST performance improves when our models are trained as an end-to-end system. This can be explained as additional supervision from system responses not only contributes to learn a good response generation network but also positively impact DST network. Additional DST results of individual domains can be seen in Appendix A.3.

For completion, we also conduct experiment for context-to-text generation setting and compare with baseline models in Appendix A.3.

**Ablation**. We experiment with different model variants in Table 3. First, we noted that removing self-attention on the joint feature domain-slot vectors ($N_{dst}^{DS} = 0$) reduces the joint accuracy performance. This self-attention is important because it allows our models to learn signals across *(domain, slot)* joint features rather than just at independently domain level and slot level. Second, ranging the number of attention layers in domain-level processing and slot-level processing from 3 to 1 gradually reduces the model performance. This shows the efficacy of our Late Fusion approach. Combining the

Table 2: Joint Accuracy metric on MultiWOZ2.1. Except for TSCP, the baseline results are as reported by Eric et al. (2019). Best results are highlighted in bold.

| **Model** | **Joint Accuracy** |
|---|---|
| HJST (Eric et al., 2019) | 35.55% |
| DST Reader (Gao et al., 2019) | 36.40% |
| TSCP (Lei et al., 2018) | 37.12% |
| FJST (Eric et al., 2019) | 38.00% |
| HyST (Goel et al., 2019) | 38.10% |
| TRADE (Wu et al., 2019a) | 45.60% |
| **Ours** | **49.55%** |

features at deeper network layers results in better joint feature representation and hence, increases the model performance. Lastly, we observed that our models can efficiently detect contextual signals from the dialogue states of previous turn *PrevBS* as the performance of our models with or without using the full dialogue history is very similar. This will benefit as the dialogue history evolves over time and our models only need to process the latest dialogue turn in combination with the predicted dialogue state in previous turn as an input.

Table 3: We experiment with several model variants by controlling following changes: whether to include dialogue state in previous turn PrevBS as an input or not, length of dialogue history (in turns) as 1 (latest turn only) or all possible dialogue turns, number of attention layers in slot-level module $N^S_{dst}$, domain-level module $N^D_{dst}$, and joint domain-slot module $N^{DS}_{dst}$. The best result in each metric is highlighted in bold.

| **PrevBS** | $length(X\_his)$ | $N^S_{dst}$ | $N^D_{dst}$ | $N^{DS}_{dst}$ | **Joint Acc** | **Slot Acc** | **Inform** | **Success** | **BLEU** |
|---|---|---|---|---|---|---|---|---|---|
| Y | 1 | 4 | 4 | 1 | **50.91%** | **97.34%** | **72.45%** | **52.14%** | 19.80 |
| Y | 1 | 4 | 4 | 0 | 47.14% | 96.92% | 71.20% | 50.40% | 19.74 |
| Y | 1 | 3 | 3 | 1 | 46.60% | 96.91% | 70.42% | 50.23% | 19.10 |
| Y | 1 | 2 | 2 | 1 | 45.88% | 96.76% | 67.23% | 50.23% | 19.15 |
| Y | 1 | 1 | 1 | 1 | 43.41% | 96.55% | 64.22% | 49.52% | 18.42 |
| Y | All | 4 | 4 | 1 | 50.21% | 97.23% | 71.42% | 51.53% | **19.85** |
| N | All | 4 | 4 | 1 | 36.32% | 92.53% | 52.13% | 35.53% | 17.31 |
| N | 1 | 4 | 4 | 1 | 29.24% | 90.04% | 42.52% | 25.32% | 15.23 |

**Qualitative Analysis**. We examine an example dialogue in the test data and compare our predicted outputs with the baseline TSCP ($L_{bspan} = 20$) (Lei et al., 2018) and the ground truth. From the table in the left of Figure 3, we observe that both our predicted dialogue state and system response are more correct than the baseline. Specifically, our dialogue state can detect the correct *type* slot in the *attraction* domain. As our dialogue state is correctly predicted, the queried results from DB is also more correct, resulting in better response with the right information (i.e. 'no attraction available'). From visualization of domain-level and slot-level attention on the user utterance, we notice important tokens of the text sequences, i.e. 'entertainment' and 'close to', are attended with higher attention scores. In addition, at domain-level attention, we find a potential additional signal from the token 'restaurant', which is also the domain from the previous dialogue turn. We also observe that attention is more refined along the neural network layers. For example, in the domain-level processing, compared to the $2^{nd}$ layer, the $4^{th}$ layer attention is more clustered around specific tokens of the user utterance. The complete predicted output for this example dialogue and other qualitative analysis can be seen in Appendix A.4.

## 5 Conclusion

In this work, we proposed an end-to-end dialogue system with a novel Multi-level Neural Belief Tracker. Our DST module can track complex belief states of multiple domains and output more accurate dialogue states. The DST is combined with attention-based generation module to generate dialogue responses. Evaluated on the large-scale multi-domain dialogue benchmark MultiWOZ2.1, our models achieve the state-of-the-art performance in DST and competitive measures in task-completion and response generation.

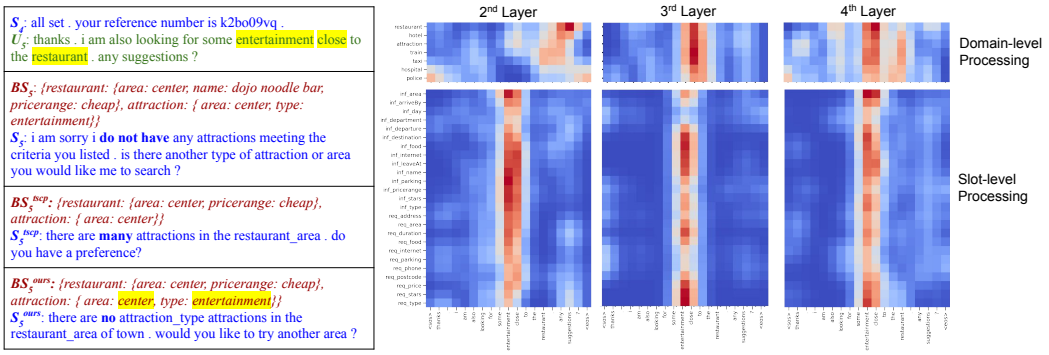


Figure 3: Example dialogue with the input system response $S_{t-1}$ and current user utterance $U_t$, and the output belief state $BS_t$ and system response $S_t$. Compared with TSCP (Row 3), our dialogue state and response (Last Row) are more correct and closer to the ground truth (Row 2). Visualization of attention to the user utterance sequence at slot-level (lower right) and domain-level (upper right) is also included. More red denotes higher attention score between domain or slot representation and token representation. Best viewed in color.

## REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. *arXiv preprint arXiv:1905.12866*, 2019.

Mihail Eric and Christopher Manning. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 468–473, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-2075.

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

Revanth Gangi Reddy, Danish Contractor, Dinesh Raghu, and Sachindra Joshi. Multi-level memory for task oriented dialogs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3744–3754, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1375. URL https://www.aclweb.org/anthology/N19-1375.

Shuyang Gao, Abhishek Sethi, Sanchit Aggarwal, Tagyoung Chung, and Dilek Hakkani-Tur. Dialog state tracking: A neural reading comprehension approach. *arXiv preprint arXiv:1908.01946*, 2019.

Rahul Goel, Shachi Paul, and Dilek Hakkani-Tür. Hyst: A hybrid approach for flexible and accurate dialogue state tracking. *arXiv preprint arXiv:1907.00883*, 2019.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2787–2792, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1300.

Homa B Hashemi, Amir Asiaee, and Reiner Kraft. Query intent detection using convolutional neural networks. In *International Conference on Web Search and Data Mining, Workshop on Query Understanding*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Matthew Henderson, Blaise Thomson, and Jason D Williams. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pp. 263–272, 2014a.

Matthew Henderson, Blaise Thomson, and Steve Young. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pp. 292–299, 2014b.

Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. Sumbt: Slot-utterance matching for universal and scalable belief tracking. *arXiv preprint arXiv:1907.07421*, 2019.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 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)*, pp. 1437–1447, 2018.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 733–743, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/I17-1074.

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

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1468–1478. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-1136.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. Structured fusion networks for dialog. *arXiv preprint arXiv:1907.10016*, 2019.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Multi-domain dialog state tracking using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 794–799, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2130. URL https://www.aclweb.org/anthology/P15-2130.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. A modular task-oriented dialogue system using a neural mixture-of-experts. *arXiv preprint arXiv:1907.05346*, 2019.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2231–2240, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1237. URL https://www.aclweb.org/anthology/D17-1237.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. Deep Dyna-Q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2182–2192, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1203.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 2, pp. 432–437, 2018.

Abhinav Rastogi, Dilek Hakkani-Tur, and Larry Heck. Scalable multi-domain dialogue state tracking. In *Proceedings of IEEE ASRU*, 2017.

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

Shang-Yu Su, Kai-Ling Lo, Yi Ting Yeh, and Yun-Nung Chen. Natural language generation by hierarchical decoding with linguistic patterns. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 61–66, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2010. URL https://www.aclweb.org/anthology/N18-2010.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3104–3112. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Lina M. Rojas Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. Conditional generation and snapshot learning in neural dialogue systems. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2153–2162, Austin, Texas, November 2016a. Association for Computational Linguistics. doi: 10.18653/v1/D16-1233. URL https://www.aclweb.org/anthology/D16-1233.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 120–129, San Diego, California, June 2016b. Association for Computational Linguistics. doi: 10.18653/v1/N16-1015. URL https://www.aclweb.org/anthology/N16-1015.

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

system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 438–449, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1042.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 808–819, Florence, Italy, July 2019a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1078.

Chien-Sheng Wu, Richard Socher, and Caiming Xiong. Global-to-local memory pointer networks for task-oriented dialogue. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019b.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1208–1218, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N19-1123.

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

# A APPENDIX

## A.1 DATA PRE-PROCESSING

First, we delexicalize each target system response sequence by replacing matched entity attribute appeared in the sequence to the canonical tag $\langle domain\_attribute \rangle$. For example, the original target response 'the train id is tr8259 departing from cambridge' is delexicalized into 'the train id is *train_id* departing from *train_departure*'. We use the provided entity databases (DBs) to match potential attributes in all target system responses. For dialogue history, we keep the original version of all text, including system responses of previous turns, rather than the delexicalized form. We split all sequences of dialogue history, user utterances of current turn, previous belief states, and delexicalized target responses, into case-insensitive tokens. We share the embedding weights of all source sequences. For source sequences, in total there are 5,491 unique tokens, including slot and domain tokens as well as $\langle eos \rangle$, $\langle sos \rangle$, $\langle pad \rangle$, and $\langle unk \rangle$ tokens. For target sequences, there are 2,648 unique tokens in total, including all canonical tags as well as $\langle eos \rangle$, $\langle sos \rangle$, $\langle pad \rangle$, and $\langle unk \rangle$ tokens. As can be seen in Table 4, in source sequences, the overlapping rates of unique tokens to the training embedding vocabulary are about 64% and 65% in validation and test set respectively. For target sequences, the overlapping rates are about 83% and 82% in validation and test set respectively.

Table 4: Numbers of unique tokens in source and target sequences.

| | **train** | **val** | **test** |
|---|---|---|---|
| ***Source Sequences*** | | | |
| **#tokens** | 15,339 | 4,831 | 4,685 |
| **#tokens with freq>1** | 5,491 | - | - |
| **#overlapping tokens with training vocabulary** | 5,491 | 3,131 | 3,047 |
| ***Target Sequences*** | | | |
| **#tokens** | 4,842 | 1,924 | 1,875 |
| **#tokens with freq>1** | 2,648 | - | - |
| **#overlapping tokens with training vocabulary** | 2,648 | 1,591 | 1,543 |

As we analyze the data, we summarize the number of dialogues in each domain in Table 5. For each domain, a dialogue is selected as long as the whole dialogue (i.e. single-domain dialogue) or parts of the dialogue (i.e. in multi-domain dialogue) is involved with the domain. For each domain, we also build a set of possible *inform* and *request* slots using the belief state annotation in the training data. The details of slots, entity attributes, and DB size, in each domain, can be seen in Table 6. The DBs of 3 domains *taxi, police*, and *hospital* were not provided in the benchmark.

Table 5: Summary of MultiWOZ dataset (Budzianowski et al., 2018) by domain

| **Domain** | **#dialogues** | | |
|---|---|---|---|
| | **train** | **val** | **test** |
| **Restaurant** | 3,817 | 438 | 437 |
| **Hotel** | 3,387 | 416 | 394 |
| **Attraction** | 2,718 | 401 | 396 |
| **Train** | 3,117 | 484 | 495 |
| **Taxi** | 1,655 | 207 | 195 |
| **Police** | 245 | 0 | 0 |
| **Hospital** | 287 | 0 | 0 |

## A.2 BASELINES

We describe a list of baseline models in DST setting and context-to-text generation setting.

### A.2.1 DIALOGUE STATE TRACKING

**FJST** and **HJST** (Eric et al., 2019). FJST and HJST follow a fixed-vocabulary approach for state tracking. Both models include encoder modules (either bidirectional LSTM or hierarchical LSTM) to encode the dialogue history. The models pass the context hidden states to separate linear transforma-

Table 6: Summary of slots and DB details by domain in the MultiWOZ dataset (Budzianowski et al., 2018)

| Domain | Slots | #entities | DB attributes |
|---|---|---|---|
| **Restaurant** | inf_area, inf_food, inf_name, inf_pricerange, inf_bookday, inf_bookpeople, inf_booktime, req_address, req_area, req_food, req_phone, req_postcode | 110 | id, address, area, food, introduction, name, phone, postcode, pricerange, signature, type |
| **Hotel** | inf_area, inf_internet, inf_name, inf_parking, inf_pricerange, inf_stars, inf_type, inf_bookday, inf_bookpeople, inf_bookstay, req_address, req_area, req_internet, req_parking, req_phone, req_postcode, req_stars, req_type | 33 | id, address, area, internet, parking, single, double, family, name, phone, postcode, pricerange', takesbookings, stars, type |
| **Attraction** | inf_area, inf_name, inf_type, req_address, req_area, req_phone, req_postcode, req_type | 79 | id, address, area, entrance, name, phone, postcode, pricerange, openhours, type |
| **Train** | inf_arriveBy, inform_day, inf_departure, inf_destination, inf_leaveAt, inf_bookpeople, req_duration, req_price | 2,828 | trainID, arriveBy, day, departure, destination, duration, leaveAt, price |
| **Taxi** | inf_arriveBy, inf_departure, inf_destination, inf_leaveAt, req_phone | - | - |
| **Police** | inf_department, req_address, req_phone, req_postcode | - | - |
| **Hospital** | req_address, req_phone, req_postcode | - | - |

tion to obtain final vectors to predict individual state slots separately. The output vector is used to measure a score of a predefined candidate set for each slot.

**TSCP** (Lei et al., 2018). TSCP is an end-to-end dialogue system that can do both DST and NLG. The model utilize pointer network to generate both dialogue states and responses. To compare with TSCP in DST setting, we adapt the model to multi-domain dialogues and report the results only on DST components. For DST experiment, we reported the performance when the maximum length of dialogue state sequence in the state decoder $L$ is set to 20 tokens.

**DST Reader** (Gao et al., 2019). This model considers the DST task as a reading comprehension task. The model predicts each slot as a span over tokens within dialogue history. DST Reader utilizes attention-based neural network with additional modules to predict slot type and slot carryover probability.

**HyST** (Goel et al., 2019). This baseline combines the advantage of both fixed-vocabulary and open-vocabulary approaches. In open-vocabulary, the set of candidates of each slot is constructed based on all word n-grams in dialogue history. Both approaches are applied in all slots and depending on their performance in validation set, the better approach is applied to predict individual slots.

**TRADE** (Wu et al., 2019a). This is the current state-of-the-art model on the MultiWOZ2.1 dataset. The model combines pointer network to generate individual slot token-by-token. The prediction is additional supported by a slot gating component that decides whether the slot is “none", “dontcare", or “pointer" (generated).

### A.2.2 CONTEXT-TO-TEXT GENERATION

Budzianowski et al. (2018) provides a baseline for this setting by following the sequence-to-sequence model (Sutskever et al., 2014) with additional signals from the belief tracker and discrete data pointer vector.

**TokenMoE** (Pei et al., 2019). TokenMoE refers to Token-level Mixture-of-Expert model. The model follows a modularized approach by separating different components known as expert bots for different dialogue scenarios. A dialogue scenario can be dependent on a domain, a type of dialogue act, etc. A chair bot is responsible for controlling expert bots to dynamically generate dialogue responses.

**HDSA** (Chen et al., 2019). This is the current state-of-the-art for context-to-text generation setting in MultiWOZ2.0. HDSA leverages the structure of dialogue acts to build a multi-layer hierarhical graph. The graph is incorporated as an inductive bias in self-attention network to improve the semantic quality of generated dialogue responses.

**Structured Fusion** (Mehri et al., 2019). This approach follows a traditional modularized dialogue system architecture, including separate components for NLU, DM, and NLG. These components are

pretrained and combined into an end-to-end system. Each component output is used as a structured input to other components.

**LaRL** (Zhao et al., 2019). This model uses a latent dialogue action framework instead of traditional handcrafted dialogue acts. The latent variables are learned using unsupervised learning with stochastic variational inference. The model are trained in a reinforcement learning framework whereby the parameters are trained to yield better Success rate.

### A.3 Additional Results

**Domain-Specific Results**. In Table 7 and 8, we presented additional results of our model and the baselines TSCP (Lei et al., 2018). For state tracking, the metrics are calculated for domain-specific slots of the corresponding domain at each dialogue turn. For task completion and response generation, we calculated the metrics for single-domain dialogues of the corresponding domain. We do not report the *Inform* metric for the *taxi* domain because no DB was provided in the benchmark for this domain. From Table 7, in each domain, our approach outperforms TSCP across most of the metrics, except the Success and BLEU metric in the *taxi* domain. In term of task-completion, our model performs better significant improvement in domains with large DB sizes such as *train* and *restaurant*. In term of response generation, our results are consistently higher than the baselines as the model can return more appropriate responses from better decoded dialogue states and DB queried results. For state tracking task alone, in Table 8, our models perform consistently in the 3 domains *attraction, restaurant*, and *train* domains. However, the performance significantly drops in the *taxi* domain. This performance drop negatively impacts the overall performance across all domains. We plan to investigate further to identify and address challenges in this particular domain in future work.

Table 7: Additional experiment results on MultiWOZ2.1. Compared to the baseline TSCP (Lei et al., 2018), our model performs better in most of the metrics in each domain. For state tracking, the metrics are calculated for domain-specific slots of the corresponding domain at each dialogue turn. For task completion and response generation, the metrics are computed for only single-domain dialogues with the corresponding domain in each row. The best result in each metric and domain is highlighted in bold.

| Model | Joint Acc | Slot Acc | Inform | Success | BLEU |
|---|---|---|---|---|---|
| *Attraction domain* | | | | | |
| TSCP (L=8) | 63.53% | 97.34% | 68.24% | 51.52% | 16.53 |
| TSCP (L=20) | 68.12% | 98.42% | 73.24% | 66.24% | 16.63 |
| **Ours** | **70.78%** | **99.06%** | **75.00%** | **66.66%** | **22.58** |
| *Hotel domain* | | | | | |
| TSCP (L=8) | 41.52% | 96.34% | 74.24% | 45.23% | 11.52 |
| TSCP (L=20) | 45.23% | 97.24% | 81.42% | 66.23% | 12.63 |
| **Ours** | **49.52%** | **97.50%** | **88.06%** | **86.57%** | **17.58** |
| *Restaurant domain* | | | | | |
| TSCP (L=8) | 64.23% | 97.34% | 91.24% | 63.23% | 15.24 |
| TSCP (L=20) | 65.13% | 98.52% | 93.23% | 78.23% | 17.52 |
| **Ours** | **66.50%** | **98.76%** | **95.16%** | **85.48%** | **20.99** |
| *Taxi domain* | | | | | |
| TSCP (L=8) | 15.52% | 91.24% | N/A | 14.32% | 7.42 |
| TSCP (L=20) | 18.92% | 94.02% | N/A | **35.23%** | **15.53** |
| **Ours** | **23.05%** | **96.42%** | N/A | 23.24% | 14.30 |
| *Train domain* | | | | | |
| TSCP (L=8) | 54.23% | 89.24% | 55.24% | 42.42% | 11.53 |
| TSCP (L=20) | 61.53% | 90.02% | 63.23% | 51.04% | 15.42 |
| **Ours** | **65.12%** | **90.22%** | **96.97%** | **87.88%** | **20.54** |

**Context-to-Text Generation**. Following Eric et al. (2019), to compare with baselines in this setting, we assumes access to the ground-truth labels of dialogue belief states and data pointer during inference. We compare with existing baselines in Table 2 (Refer to Appendix A.2 for more description of the baselines). Our model achieves the state-of-the-art in the Inform metric but do not perform as well in terms of Success metric. We achieve a competitive BLEU score, only behind the current state-of-the-art HDSA model. An explanation for our model not able to achieve a high Success metric is that we did not utilize the dialogue act information. The current state-of-the-art HDSA leverages the graph structure of dialogue acts into dialogue models. Furthermore, compared to approaches such

Table 8: Additional experiment results of state tracking on MultiWOZ2.1. In each domain, the metrics are calculated for domain-specific slots of the corresponding domain at each dialogue turn. The best result in each domain is highlighted in bold.

| Domain | Joint Acc | Slot Acc |
|---|---|---|
| Attraction | 67.91% | 98.97% |
| Hotel | 47.48% | 97.58% |
| Restaurant | 65.39% | 98.62% |
| Taxi | 23.05% | 96.35% |
| Train | 66.00% | 98.24% |

Table 9: Performance for context-to-text generation setting on MultiWOZ2.0. The baseline results are as reported in the benchmark leaderboard.

| Model | Inform | Success | BLEU |
|---|---|---|---|
| Baseline (Budzianowski et al., 2018) | 71.29% | 60.96% | 18.80 |
| TokenMoE (Pei et al., 2019) | 75.30% | 59.70% | 16.81 |
| HDSA (Chen et al., 2019) | 82.90% | 68.90% | **23.60** |
| Structured Fusion (Mehri et al., 2019) | 82.70% | 72.10% | 16.34 |
| LaRL (Zhao et al., 2019) | 82.78% | **79.20%** | 12.80 |
| **Ours** | **83.83%** | 67.36% | 19.88 |

as (Pei et al., 2019; Mehri et al., 2019), our model does not use pretrained network modules such as NLU and DST. Our end-to-end setting is more related along the line of research work for end-to-end dialogue systems without relying on system action annotation (Lei et al., 2018; Madotto et al., 2018; Wu et al., 2019b). To improve the Success metric, we plan to extend our work in the future that can derive better dialogue policy for higher Success rate.

### A.4 Additional Qualitative Analysis

In Table 10, we reported the complete output of an example multi-domain dialogue. Overall, our dialogue agent can carry a proper dialogue with the user throughout the dialogue steps. Specifically, we observed that our model can detect new domains at dialogue steps where the domains are introduced e.g. *attraction* domain at the $5^{th}$ turn and *taxi* domain at the $8^{th}$ turn. The dialogue agent can also detect some of the co-references among the domains. For example, at the $5^{th}$ turn, the dialogue agent can infer the slot *area* for the new domain *attraction* as the user mentioned 'close the restaurant'. We noticed that that at later dialogue steps such as the $6^{th}$ turn, our decoded dialogue state is not correct possibly due to the incorrect decoded dialogue state in the previous turn, i.e. $5^{th}$ turn.

In Figure 4 and 5, we reported the Joint Goal Accuracy and BLEU metrics of our model by dialogue turn. As we expected, the Joint Accuracy metric tends to decrease as the dialogue history extends over time. The dialogue agent achieves the highest accuracy in state tracking at the $1^{st}$ turn and gradually reduces to zero accuracy at later dialogue steps, i.e. $15^{th}$ to $18^{th}$ turns. For response generation performance, the trend of BLEU score is less obvious. The dialogue agent obtains the highest BLEU scores at the $3^{rd}$ turn and fluctuates between the $2^{nd}$ and $13^{th}$ turn.

Table 10: Complete results of an example multi-domain dialogue, including the input of past system response $S_{t-1}$ and current user utterance $U_t$, and the predicted dialogue belief state $BS_t$ and system response $S_t$. The dialogue includes 11 turns in total and extends across 3 domains sequentially: restaurant, attraction, and taxi.

| Turn | Input Dialogue | Ground-truth State and Response | Our Predicted State and Response |
|---|---|---|---|
| | S0: <blank> | BS1: {restaurant: {area: centre, pricerange: cheap}} | BS1: {restaurant: {area: centre, pricerange: cheap}} |
| 1 | U1: i am looking for a cheap restaurant in the center of the city | S1: do you have any specific type of food you would like ? | S1: there are 15 restaurant_pricerange restaurant_type s in the restaurant_area of town . do you have a preference ? |
| | S1: do you have any specific type of food you would like ? | BS2: {restaurant: {area: centre, pricerange: cheap}} | BS2: {restaurant: {area: centre, pricerange: cheap}} |
| 2 | U2: no , i am not picky as long as the price -s are low . | S2: there is a restaurant_pricerange restaurant_food restaurant_type called the restaurant_name located in the restaurant_area of town . would you like to book a table ? | S2: how about restaurant_name ? it s an restaurant_pricerange restaurant_food restaurant_type located in the restaurant_area of town . |
| | S2: there is a cheap chinese restaurant called the dojo noodle bar located in the centre of town . would you like to book a table ? | BS3: {restaurant: {area: centre, pricerange: cheap, name: dojo noodle bar}} | BS2: {restaurant: {area: centre, pricerange: cheap}} |
| 3 | U3: yes please , for 8 people at 1830 on thursday . | S3: i am sorry but restaurant_name is solidly booked at that time . i can try a different time or day for you . | S3: booking was successful . the table will be reserved for 15 minutes . reference number is restaurant_reference . |
| | S3: i am sorry but dojo noodle bar is solidly booked at that time . i can try a different time or day for you | BS4: {restaurant: {area: centre, pricerange: cheap, name: dojo noodle bar}} | BS4: {restaurant: {area: centre, pricerange: cheap}} |
| 4 | U4: can you try to book it at 1730 . | S4: all set . your reference number is restaurant_reference . | S4: booking was successful . the table will be reserved for 15 minutes . reference number is restaurant_reference . |
| | S4: all set . your reference number is k2bo09vq . | BS5: {restaurant: {area: centre, pricerange: cheap, name: dojo noodle bar}, attraction: {area: centre, type: entertainment}} | BS5: {restaurant: {area: centre, pricerange: cheap}, attraction: {area: centre, type: entertainment}} |
| 5 | U5: thanks . i am also looking for some entertainment close to the restaurant . any suggestions ? | S5: 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 ? | S5: there are no attraction_type attractions in the restaurant_area of town . would you like to try another area ? |
| | S5: 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 ? | BS6:{restaurant: {area: centre, pricerange: cheap, name: dojo noodle bar}, attraction: {area: centre, type: architecture}} | BS6:{restaurant: {area: centre, pricerange: cheap}, attraction: {area: centre, type: architecture}} |
| 6 | U6: why do not you try an architectural attraction , instead . | S6: attraction_name looks good , would you like to head there ? | S6: attraction_name is a great attraction_type in the center of town . would you like more information ? |
| | S6: all saints church looks good , would you like to head there ? | BS7:{restaurant: {area: centre, pricerange: cheap, name: dojo noodle bar}, attraction: {area: centre, type: architecture}} | BS7:{restaurant: {area: centre, pricerange: cheap}, attraction: {area: centre, type: architecture}} |
| 7 | U7: yes please . can you let me know what the entrance fee is too ? | S7: its entrance fee is attraction_pricerange . | S7: the entrance fee is attraction_pricerange . is there anything else i can help you with ? |
| | S7: its entrance fee is free . | BS8:{restaurant: {area: centre, pricerange: cheap, name: dojo noodle bar}, attraction: {area: centre, type: architecture}, taxi: {departure: dojo noodle bar, destination: all saints church}} | BS8:{restaurant: {area: centre, pricerange: cheap}, attraction: {area: centre, type: architecture}, taxi: {destination: dojo noodle bar}} |
| 8 | U8: i also need to book a taxi between the restaurant and the church . | S8: what time would you like the taxi to pick you up from restaurant_name ? | S8: what time would you like to leave or arrive by ? |
| | S8: what time would you like the taxi to pick you up from dojo noodle bar ? | BS9: {restaurant: {area: centre, pricerange: cheap, name: dojo noodle bar}, attraction: {area: centre, type: architecture}, taxi: {departure: dojo noodle bar, destination: all saints church, leaveAt: 1830}} | BS9:{restaurant: {area: centre, pricerange: cheap}, attraction: {area: centre, type: architecture}, taxi: {destination: dojo noodle bar}} |
| 9 | U9: 1830 , please . | S9: your taxi is booked . the car is a taxi_type and the contact number is taxi_phone . is there anything else i can help you with ? | S9: booking completed ! booked car type taxi_type contact number taxi_phone can i help you with anything else ? |
| | S9: your taxi is booked . the car is a blue bmw and the contact number is 07239644669 . is there anything else i can help you with ? | BS10: {restaurant: {area: centre, pricerange: cheap, name: dojo noodle bar}, attraction: {area: centre, type: architecture}, taxi: {departure: dojo noodle bar, destination: all saints church, leaveAt: 1830}} | BS10: {restaurant: {area: centre, pricerange: cheap}, attraction: {area: centre, type: architecture}, taxi: {destination: dojo noodle bar}} |
| 10 | U10: that s it . thanks . | S10: awesome , enjoy your trip ! | S10: you are welcome . have a great day ! |
| | S10: awesome , enjoy your trip ! | BS11: {restaurant: {area: centre, pricerange: cheap, name: dojo noodle bar}, attraction: {area: centre, type: architecture}, taxi: {departure: dojo noodle bar, destination: all saints church, leaveAt: 1830}} | BS11: {restaurant: {area: centre, pricerange: cheap}, attraction: {area: centre, type: architecture}, taxi: {destination: dojo noodle bar}} |
| 11 | U11: thanks ! i will definitely try to . | S11: great ! goodbye and have a nice day . | S11: thank you for using our services . have a great day . |

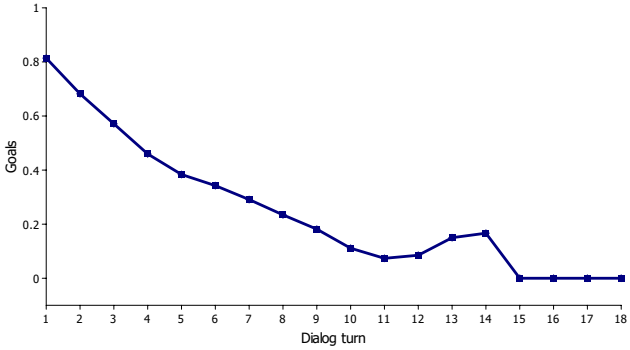


Figure 4: Joint Accuracy metric by dialogue turn in the test data.

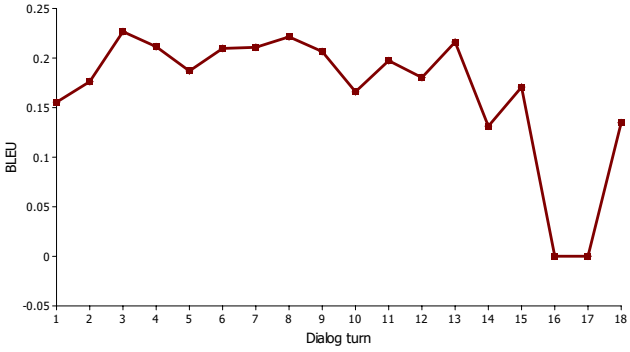


Figure 5: BLEU4 metric by dialogue turn in the test data.