MoNET: Tackle State Momentum via Noise-Enhanced Training for Dialogue State Tracking

Haoning Zhang[1,3], **Junwei Bao**[2*], **Haipeng Sun**[2],
Youzheng Wu[2], **Wenye Li**[4,5], **Shuguang Cui**[3,1,6], **Xiaodong He**[2]
[1]FNii, CUHK-Shenzhen [2]JD AI Research
[3]SSE, CUHK-Shenzhen [4]SDS, CUHK-Shenzhen [5]SRIBD [6]Pengcheng Lab
haoningzhang@link.cuhk.edu.cn, {wyli, shuguangcui}@cuhk.edu.cn,
{baojunwei, sunhaipeng6, wuyouzheng1, xiaodong.he}@jd.com

Abstract

Dialogue state tracking (DST) aims to convert the dialogue history into dialogue states which consist of slot-value pairs. As condensed structural information memorizes all dialogue history, the dialogue state in the previous turn is typically adopted as the input for predicting the current state by DST models. However, these models tend to keep the predicted slot values unchanged, which is defined as *state momentum* in this paper. Specifically, the models struggle to *update* slot values that need to be changed and *correct* wrongly predicted slot values in the previous turn. To this end, we propose MoNET to tackle state momentum via noise-enhanced training. First, the previous state of each turn in the training data is noised via replacing some of its slot values. Then, the noised previous state is used as the input to learn to predict the current state, improving the model's ability to *update* and *correct* slot values. Furthermore, a contrastive context matching framework is designed to narrow the representation distance between a state and its corresponding noised variant, which reduces the impact of noised state and makes the model better understand the dialogue history. Experimental results on MultiWOZ datasets show that MoNET outperforms previous DST methods. Ablations and analysis verify the effectiveness of MoNET in alleviating state momentum issues and improving the anti-noise ability[1].

1 Introduction

Dialogue state tracking (DST) is a core component in modular task-oriented dialogue systems (Hosseini-Asl et al., 2020; Yang et al., 2021; Sun et al., 2022, 2023). It extracts users' intents from the dialogue history and converts them into structural dialogue states, i.e., sets of slot-value pairs. An accurate dialogue state is crucial for generating correct dialogue action and suitable natural language responses, which are the main tasks of dialogue management and natural language generation components (Williams and Young, 2007; Thomson and Young, 2010; Young et al., 2010). Earlier DST approaches predict the state directly from the dialogue history (natural language utterances) (Mrkšić et al., 2017; Xu and Hu, 2018; Wu et al., 2019; Chen et al., 2020a). Since the dialogue state is condensed structural information memorizing all dialogue history, recent methods incorporate the previously predicted state as the input besides the dialogue history (Ouyang et al., 2020; Kim et al., 2020; Ye et al., 2021).

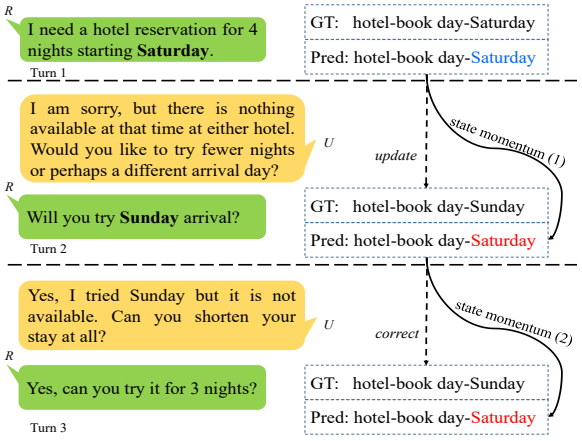

Figure 1: A dialogue example of three turns, containing the system utterance (*U*), the user response (*R*), the ground truth dialogue state (GT), and the prediction of each turn (Pred). The state "hotel-book day-Saturday" is predicted in the first turn (marked in blue). The dotted arrow represents the ideal predictions, i.e., *update* slot values that need to be changed (Turn 2) and *correct* wrongly predicted slot values in the previous turn (Turn 3). The solid arrow represents the predictions (marked in red) with state momentum issues.

*Corresponding author: baojunwei001@gmail.com
[1]Our code is available at https://github.com/JD-AI-Research-NLP/MoNET

Findings of the Association for Computational Linguistics: ACL 2023, pages 520–534
July 9-14, 2023

Conventional DST models taking the previous state as the input usually show the characteristic that the previously predicted slot values tend to be kept unchanged when predicting the current state, defined as *state momentum* in this paper. The state momentum makes DST models struggle to modify the previous prediction, which affects the performance when the values of some slots need to be *updated* as the user's intent changes, and there exist wrongly predicted slot values that need to be *corrected*. Figure 1 gives an example of a dialogue involving three turns with two types of state momentum issues. The state **hotel-book day-Saturday** is predicted in Turn 1 and keeps unchanged in the next two turns, while the user's request is updated into **Sunday** in Turn 2. Consequently, the predicted state becomes wrong in the following two turns. The dotted arrow represents the ideal prediction cases: the value is *updated* with the ground truth changes and is *corrected* when becoming a wrong input. The solid arrow represents the state momentum issues, where the state is kept unchanged, leading to two consecutive wrong predictions. One possible reason for the state momentum issue is that in the training data, most slot values in the previous turn are the same as those in the current turn, which limits the ability of conventional DST models to modify slot values during inference. To address this limitation, an intuitive idea is to augment training instances with a higher ratio of slots whose previous values differ from those in the current turn. By incorporating such examples, the DST model can learn to deal with more cases where modifying previous predictions is required. Besides, if the DST model can treat wrong and correct dialogue states similarly in representations, then the former will typically help make further predictions. In other words, by treating incorrect dialogue states as valuable information, the DST model can potentially identify and correct erroneous slot values.

In this paper, we propose **MoNET** to tackle the state momentum issue via a noise-enhanced training strategy. The core idea is to manually add noise into the previous state to simulate scenarios with wrong state input. First, the previous state of each turn in the training data is noised via replacing some of its slot values. Specifically, for each active slot (with a non-*none* value), we replace its value with a certain probability. Then, the noised previous state, concatenated with the dialogue history, is used as the input to learn to predict the current state, improving the model's ability to *update* and *correct* slot values. Furthermore, a contrastive context matching framework is designed to narrow the representation distance between a state and its corresponding noised variant, which reduces the impact of the noised state and makes the DST model better understand the dialogue history. Such approaches make the model less sensitive to the noise, and enhance its ability to modify the slot values of previous states in current predictions. Experiments on the multi-domain dialogue datasets MultiWOZ 2.0, 2.1, and 2.4 show that our MoNET outperforms previous DST models. Ablation studies and analysis further verify the effectiveness of the proposed noised DST training and the contrastive context matching framework in alleviating state momentum and improving the model's anti-noise ability.

The contributions are summarized as follows: (1) We define the state momentum issue in DST, where models tend to keep the predicted slot values unchanged, namely, struggling to *update* and *correct* them from the previous turn. (2) We propose MoNET to tackle the state momentum issue via noised DST training and the contrastive context matching framework. (3) We conduct comprehensive experiments on three datasets, MultiWOZ 2.0, 2.1, and 2.4. The results demonstrate that MoNET outperforms previous DST methods, showcasing its effectiveness in alleviating the state momentum issue.

2 Related Work

2.1 Dialogue State Tracking

Traditional DST approaches focus on single-domain dialogue state tracking (Williams and Young, 2007; Thomson and Young, 2010; Lee and Kim, 2016). Recent researches pay more attention to multi-domain DST using distributed representation learning (Wen et al., 2017; Mrkšić et al., 2017). Previous works implement Seq2seq frameworks to encode the dialogue history, then predict the dialogue state from scratch at every turn (Rastogi et al., 2017; Ren et al., 2018; Lee et al., 2019; Wu et al., 2019; Chen et al., 2020a). Utilizing dialogue history is limited for larger turns, since the state of each turn is accumulated from all previous turns, while it's hard to retrieve state information from a long history.

Current works mainly incorporate the previous state as the model input, which is regarded as an explicit fixed-sized memory (Ouyang et al., 2020; Ye et al., 2022a; Wang et al., 2022). Kim et al. (2020) propose a state operation sub-task, where the model is trained to first predict the operation of each slot-value pair, such as UPDATE, CARRYOVER, etc., then only the value of a minimal subset of slots will be newly modified (Zeng and Nie, 2020; Zhu et al., 2020). These methods enhance model prediction efficiency and the ability to *update* slot-value pairs. Tian et al. (2021) deal with the error propagation problem that mistakes are prone to be carried over to the next turn, and design a two-pass generation process, where a temporary state is first predicted then used to predict the final state, enhancing the ability to *correct* wrong predictions. In this paper, we use "state momentum" to define the issue where the wrong dialogue state is predicted due to that the previous prediction keeps unchanged, either it should be *updated* or *corrected*. To the best of our knowledge, this is the first time to systematically tackle the issue caused by continuous unchanged predictions in the multi-turn DST task.

2.2 Contrastive Learning

Contrastive learning aims to generate high-quality representations by constructing pairs of similar examples to learning semantic similarity (Mnih and Teh, 2012; Baltescu and Blunsom, 2015; Peters et al., 2018). The goal is to help the model semantically group similar instances together and separate dissimilar instances. During training, the neighbors with similar semantic representations (**positive pairs**) will be gathered, while the non-neighbors (**negative pairs**) will be pushed apart, enabling the learning of more meaningful representations. In the NLP area, semantic representations can be learned through self-supervised methods, such as center word prediction in Word2Vec, next sentence prediction in BERT, sentence permutation in BART, etc (Mikolov et al., 2013; Devlin et al., 2019; Lewis et al., 2020). Recent approaches build augmented data samples through token shuffling, word deletion, dropout, and other operations (Cai et al., 2020; Klein and Nabi, 2020; Yan et al., 2021; Wang et al., 2021; Gao et al., 2021; Zhang et al., 2022). In this paper, we construct augmented samples based on the noised and original dialogue state. Given context inputs with the same dialogue history and different states, the model is trained to gather them into similar objects, aiming to learn better representations, reduce the impact of noise, and better understand the dialogue history.

3 Methodology

3.1 Problem Formulation

In this paper, we focus on building a dialogue state tracking (DST) model which accurately predicts the dialogue state based on the dialogue history and the previous state during multi-turn dialogue interactions. A dialogue state consists of domain-slot-value tuples, typically corresponding to the dialogue topic, the user's goal, and the user's intent. Following previous studies, in the rest of this paper, we omit "domain" and use "slot" to refer to a "domain-slot" pair. All slot-value pairs are from a pre-defined ontology.

Formally, let's define $D_t = [U_t, R_t]$ as a pair of system utterance U_t and user response R_t in the t-th turn of a multi-turn dialogue, and B_t as the corresponding dialogue state. Each state B_t contains a set of slot-value pairs, i.e., $B_t = \{(S_j, V_j^i) | j \in [1:J]\}$, where J is the total number of slots, and $V_j^i \in \mathcal{V}_j$ is one of the values in $\mathcal{V}_j$ for the j-th slot S_j in the ontology. Given the dialogue history $\{D_1, ..., D_t\}$ and previous state B_{t-1}, the goal of the DST task is to predict the current dialogue state B_t.

3.2 MoNET

As introduced in Section 1, solving the *state momentum* issue is crucial to the DST task. Therefore, in this paper, we propose MoNET to tackle the state momentum issue via a noise-enhanced training strategy to enhance the model's ability to *update* and *correct* slot values. The architecture of MoNET is shown in Figure 2(a), which consists of context BERT encoders, slot and value BERT encoders, the slot-context attention module, the slot-value matching module, and the contrastive context matching framework. Each of them will be elaborated on in this section.

3.2.1 Base Architecture

We first introduce the base architecture of our MoNET, similar to the backbone model in (Ye et al., 2022a). A model trained only with the base architecture of MoNET is noted as "Baseline", and

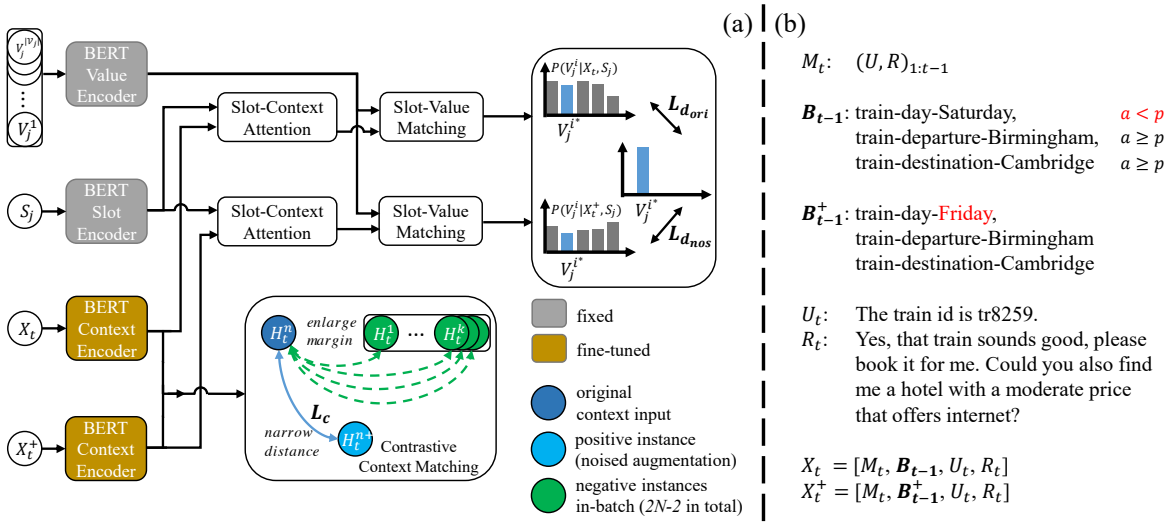

Figure 2: The model description and noised input construction example. The left part (a) shows the architecture of the MoNET model. A context input representation H^n_t in an N-size batch is shown in the contrastive context matching framework as an example, where H^{n+}_t is the context representation of its noised variant. The right part (b) gives an example of constructing noised context input. For each active slot in state B_{t-1}, given a noise threshold p, a random number a is selected. If $a < p$, then its value is replaced with another one randomly selected from the ontology (suppose the pair "train-day-Saturday" is replaced with "train-day-Friday"); otherwise the value will be kept unchanged (suppose the pairs "train-departure-Birmingham" and "train-destination-Cambridge" are unchanged).

evaluated in Section 5 to compare the difference in performance with the whole MoNET model.

Context Encoder. A BERT encoder encodes the context input, which is the concatenation of the dialogue history and the state in the previous turn:

$$\begin{aligned} X_t &= f(M_t, B_{t-1}, D_t) \\ &= [CLS] \oplus M_t \oplus B_{t-1} \oplus [SEP] \oplus D_t \oplus [SEP], \end{aligned} \tag{1}$$

where $M_t = D_1 \oplus, ..., \oplus D_{t-1}$ contains previous utterances, B_{t-1} is the state containing the active slots in the previous turn, $[CLS]$ and $[SEP]$ are special tokens of the BERT encoder. Then the representations of the context input are derived:

$$H_t = BERT(X_t) \in \mathbb{R}^{|X_t| \times d}, \tag{2}$$

where $|X_t|$ is the total number of tokens in X_t, and d is the encoded hidden size.

Slot and Value Encoders. The BERT encoders with fixed parameters are used to derive the slot and value representations:

$$\begin{aligned} h_{S_j} &= BERT_{fixed}(S_j)_{[CLS]}, \\ h_{V^i_j} &= BERT_{fixed}(V^i_j)_{[CLS]}, \end{aligned} \tag{3}$$

where states $h_{S_j}, h_{V^i_j} \in \mathbb{R}^{1 \times d}$ are the $[CLS]$ representations of the slot and value.

Slot-Context Attention. For each slot S_j, its slot-context-specific feature is extracted by the multi-head attention mechanism (Vaswani et al., 2017):

$$r^t_{S_j} = LN(MultiHead(h_{S_j}, H_t, H_t)) \in \mathbb{R}^{1 \times d}, \tag{4}$$

where LN is the normalization layer.

Slot-Value Matching. The probability of predicting the value V^i_j of the slot S_j is derived by calculating the $L2$-distance between the value representation $h_{V^i_j}$ and the slot-context representation $r^t_{S_j}$, which is denoted as:

$$P_\theta(V^i_j | X_t, S_j) = \frac{exp(-||r^t_{S_j} - h_{V^i_j}||_2)}{\sum_{k \in [1:|\mathcal{V}_j|]} exp(-||r^t_{S_j} - h_{V^k_j}||_2)}, \tag{5}$$

where θ are trainable parameters of the model.

Training and Inference. During training, the ground dialogue state is used to form the context input X_t (teacher-forcing). For the t-th turn, the loss is the sum of the negative log-likelihood among all J slots as follows:

$$L_{d_{ori}} = \sum_{j=1}^{J} -log(P_\theta(V^{i^*}_j | X_t, S_j)), \tag{6}$$

where $V_j^{i^*}$ is the ground truth value of the slot S_j at turn t. During inference, the previously predicted state is used to form the context input X_t, and the value of the slot S_j is predicted by selecting the one with the smallest distance, corresponding to the largest probability:

$$\widehat{V_j^i} = \underset{i \in [1:|\mathcal{V}_j|]}{argmax}\, P_\theta(V_j^i | X_t, S_j). \quad (7)$$

3.2.2 Noised Data Construction

As described previously, an intuitive idea to tackle the state momentum issue is to increase the number of training instances where the slot-value pairs in the previous turn are different from those in the current turn. Based on this point, we attempt to utilize noised data to train the DST model.

Generally, for each active slot (with a non-*none* value) in the previous dialogue state, we involve noise by replacing its original value with another value with a probability p (used as the noise threshold), e.g., as the example shown in Figure 2(b). Formally, at each training step, given a batch of training instances, a noised context input X_t^+ is constructed for each instance based on its original context input $X_t = f(M_t, B_{t-1}, D_t)$ as follows:

$$\begin{aligned} X_t^+ &= f(M_t, B_{t-1}^+, D_t), \\ B_{t-1}^+ &= \{(S_j, V_j^{i^+}) | j \in [1:J]\}. \end{aligned} \quad (8)$$

For each active slot S_j in $B_{t-1} = \{(S_j, V_j^i)\}$, a real number $a \in [0, 1]$ is sampled to determine whether the original V_j^i is replaced with a randomly selected value $V_j^k \in \mathcal{V}_j \setminus \{V_j^i\}$ from the ontology or kept unchanged:

$$V_j^{i^+} = \begin{cases} V_j^k, & if \quad a < p \\ V_j^i, & if \quad a \geq p. \end{cases} \quad (9)$$

3.2.3 Noised State Tracking

Similar to X_t, the noised context instance X_t^+ is also used as the model input to predict the state B_t as the training target, aiming to improve the model's ability to dynamically modify the previous slot values in current predictions. Specifically, the representation H_t^+ of X_t^+ is first derived by the BERT context encoder mentioned in Section 3.2.1:

$$H_t^+ = BERT(X_t^+) \in \mathbb{R}^{|X_t^+| \times d}. \quad (10)$$

Then, similar to the previous process, for each slot S_j, X_t^+ is used to predict its value based on the distribution $P_\theta(V_j^i | X_t^+, S_j)$. Eventually, the loss for the noised state tracking can be denoted as:

$$L_{d_{nos}} = \sum_{j=1}^{J} -log(P_\theta(V_j^{i^*} | X_t^+, S_j)). \quad (11)$$

3.2.4 Contrastive Context Matching

Inspired by contrastive learning approaches which group similar samples closer and diverse samples far from each other, a contrastive context matching framework is designed to narrow the representation distance between X_t and its noised variant X_t^+, aiming to reduce the impact of the noised state B_{t-1}^+ and help the model better understand the dialogue history. Specifically, in a batch of N instances with the original context input $\mathbf{X}_t = \{X_t^n\}_{n=1}^N$, we construct N corresponding noised instances with the context input $\mathbf{X}_t^+ = \{X_t^{n+}\}_{n=1}^N$. To clearly describe the context inputs, in this section, we temporarily involve n into X_t & H_t as X_t^n & H_t^n to indicate the in-batch index.

For each context input X_t^n, its noised sample X_t^{n+} is regarded as its positive pair, and the rest $(2N-2)$ instances in the same batch with different dialogue histories are considered negative pairs. Then the model is trained to narrow the distance of the positive pair and enlarge the distance of negative pairs in the representation space with the following training objective:

$$L_c = -log(\frac{exp(sim(H_t^{n[cls]}, H_t^{n+[cls]})/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq n]} exp(sim(H_t^{n[cls]}, H_t^{k[cls]})/\tau)}), \quad (12)$$

where $H_t^{n[cls]}$ and $H_t^{n+[cls]}$ are the $[CLS]$ representations of H_t^n and H_t^{n+}, τ is the temperature parameter, and $sim(\cdot)$ indicates the cosine similarity function (Chen et al., 2020c).

3.2.5 Optimization

The total training loss for each instance is the sum of losses from the slot-value matching for DST and the contrastive context matching for representation learning, where the former is the average of the losses using the original or the noised context input mentioned in Section 3.2.1 and 3.2.3:

$$L_{tot} = (L_{d_{ori}} + L_{d_{nos}})/2 + L_c. \quad (13)$$

4 Experiment Setting

4.1 Datasets

We choose MultiWOZ, 2.0, 2.1, and 2.4 versions as our datasets. MultiWOZ 2.0 (Budzianowski et al.,

Baseline	Pre-trained Model	MultiWOZ 2.0		MultiWOZ 2.1		MultiWOZ 2.4	
		Joint	Slot	Joint	Slot	Joint	Slot
TRADE (Wu et al., 2019)	-	48.62	96.92	45.60	96.55	55.05	97.62
SUMBT (Lee et al., 2019)	BERT-base	42.40	-	49.01	96.76	61.86	97.90
PIN (Chen et al., 2020a)	-	52.44	97.28	48.40	97.02	58.92	98.02
SOM-DST (Kim et al., 2020)	BERT-base	51.72	-	53.01	-	66.78	98.38
CSFN-DST (Zhu et al., 2020)	BERT-base	52.23	-	53.19	-	-	-
DST-Picklist (Zhang et al., 2020)	BERT-base	54.39	-	53.30	97.40	-	-
SAVN (Wang et al., 2020)	BERT-base	54.52	97.42	54.86	97.55	60.55	98.05
SST (Chen et al., 2020b)	BERT-base	51.17	-	55.23	-	-	-
SimpleTOD (Hosseini-Asl et al., 2020)	DistilGPT2	-	-	55.26	-	-	-
Seq2SeqDU (Feng et al., 2021)	BERT-base	-	-	56.10	-	-	-
STAR (Ye et al., 2021)	BERT-base	54.53	-	56.36	97.59	73.62	98.85
SDP-DST (Lee et al., 2021)	T5-base	-	-	56.66	-	-	-
DS-Graph (Lin et al., 2021)	GPT2	54.86	97.47	-	-	-	-
DSGFNet (Feng et al., 2022)	BERT-base	-	-	56.70	-	-	-
PPTOD (Su et al., 2022)	T5-large	53.89	-	57.45	-	-	-
Baseline	BERT-base	54.38	97.47	55.82	97.51	73.81	98.82
MoNET	BERT-base	**55.48** (↑ 1.10)	97.55	**57.71** (↑ 1.89)	97.71	**76.02** (↑ 2.21)	98.99
Use Modified Label							
TripPy (Heck et al., 2020)	BERT-base	-	-	55.29	-	-	-
TripPy + SCoRe (Yu et al., 2021)	BERT-base	-	-	60.48	-	-	-
TripPy + CoCoAug (Li et al., 2021)	BERT-base	-	-	60.53	-	-	-
TripPy + SaCLog (Dai et al., 2021)	BERT-base	-	-	60.61	-	-	-

Table 1: Joint and slot goal accuracy of our MoNET and several previous methods on three MultiWOZ test sets.

2018) is a standard human-human conversational dialogue corpus with seven domains. MultiWOZ 2.1 (Eric et al., 2020) has the same dialogues as the 2.0 version, where some incorrect state labels are re-annotated. Both of them are widely used in previous DST approaches. MultiWOZ 2.4 (Ye et al., 2022b) is the latest refined version correcting all the incorrect state labels in validation and test sets. All three datasets contain the same number of dialogues, which are 8438/1000/1000 in train/validation/test sets. For the three datasets, we follow the previous work (Wu et al., 2019) to use five domains (attraction, hotel, restaurant, taxi, train) with 30 domain-slot pairs in experiments, since the dialogues in the remaining domains are not in the validation and test sets.

4.2 Evaluation Metrics

We use joint and slot goal accuracy as the evaluation metrics. Joint goal accuracy is the ratio of dialogue turns where the values of all slots are correctly predicted. Slot goal accuracy is the ratio of domain-slot pairs whose values are correctly predicted. Both of them include correctly predicting those inactive slots with the value $none$.

4.3 Existing Methods

We compare the performance of our MoNET with several existing methods, i.e., TRADE, SUMBT, PIN, SOM-DST, CSFN-DST, DST-Picklist, SAVN, SST, SimpleTOD, TripPy, STAR, SDP-DST, DS-Graph, DSGFNet, PPTOD shown in Table 1, and our base architecture mentioned in Section 3.2.1, denoted as Baseline.

4.4 Training Details

The BERT-base-uncased model is used as the context, slot and value encoders, with 12 attention layers and a hidden size of 768. During training, only the parameters of the context BERT encoder are updated, while the parameters of the slot and value BERT encoders are not. The batch size is set to 8. The AdamW optimizer is applied to optimize the model with the learning rate 4e-5 and 1e-4 for the context encoders and the remaining modules, respectively (Loshchilov and Hutter, 2019). The temperature parameter τ is set to 0.1. The noise threshold p defined in Section 3.2.2 is set to 0.3, and its impact on model performance is discussed in Section 5. All models are trained on a P40 GPU device for 6-8 hours.

5 Results and Analysis

5.1 Main Results

Table 1 shows performances of MoNET and baselines on MultiWOZ 2.0, 2.1 and 2.4. Among them, TripPy and its modified versions employ a ground truth label map of synonyms replacement as extra supervision, which increases their accuracy scores and differs from other methods of testing with common labels. As can be observed, MoNET achieves the joint goal accuracy scores of

Model	NoisedCM	NoisedST	Accuracy
Baseline w/o state	×	×	64.94
Baseline	×	×	73.81
MoNET-ST	×	✓	75.54
MoNET-CM	✓	×	75.76
MoNET	✓	✓	76.02

Table 2: Joint goal accuracy on the MultiWOZ 2.4 test set of MoNET and four ablated modifications.

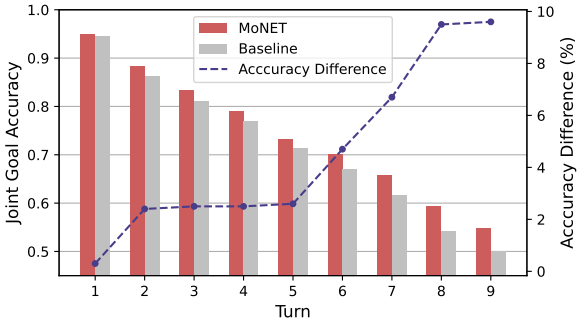

Figure 3: Turn-level joint goal accuracy and accuracy difference between MoNET and Baseline on the MultiWOZ 2.4 test set.

55.48%, 57.71%, 76.02% in three datasets, which are impressive results compared with previous methods, and has improvements of 1.10%, 1.89%, and 2.21% on the Baseline model, indicating that our proposed noise-enhanced training helps the model make better predictions.

Besides the general joint and slot goal accuracy, we also calculate the slot-level proportion of state momentum errors over all wrong predictions. We train the Baseline model and make predictions on the MultiWOZ 2.4 test set. For each dialogue, starting from the second turn, we add up each wrong predicted slot-value pair which also exists in the previous turn. Finally, there are 844 such wrong slot-value pairs, and the number of all the wrong predicted pairs is 2603, hence the proportion is (844/2603)*100%=32.4%, and our MoNET model modifies 47.0% of them (397 in 844 are correctly predicted). Moreover, in MultiWOZ 2.4 training set annotations, for each dialogue turn (also except the first turn of each dialogue), around 78.1% slot-value pairs exist in the previous turn, since the slot-value pairs will be accumulated as the dialogue progresses. The results further indicate the issue caused by those unchanged slot-value pairs during multi-turn interactions, and the effectiveness of our method in enhancing the model's ability to modify previous predictions.

5.2 Ablation Study

To explore the individual contribution of each part of our model, we compare the whole MoNET with several ablated versions. First, we remove the previous dialogue state from the context input of the Baseline model, where the modified context input is $X_t = [CLS] \oplus M_t \oplus [SEP] \oplus D_t \oplus [SEP]$, denoted as Baseline w/o state; besides, the two noise-enhanced methods are removed from the MoNET respectively, denoted as MoNET-CM (context matching only) and MoNET-ST (noised state tracking only).

Table 2 shows the joint goal accuracy performances of the full MoNET model and its four modifications on the MultiWOZ 2.4 test set. As can be observed, Baseline w/o state gets the lowest accuracy, demonstrating that explicitly using the previous dialogue state as part of the model input is beneficial to make predictions, even though there may exist wrong slot-value pairs. Besides, both MoNET-CM and MoNET-ST outperform the Baseline model, demonstrating functionalities of the noised state tracking in modifying slot-value pairs in further turns, and the context matching framework in learning improved semantic representations. Moreover, MoNET derives the best performance, demonstrating the effectiveness of integrating the two parts into a unified noised-enhance training strategy.

5.3 Turn-Level Evaluation

Figure 3 shows the turn-level joint goal accuracy of MoNET and Baseline models, as well as the percentage difference in accuracy (the difference between the two models' accuracy divided by the accuracy of Baseline) on the MultiWOZ 2.4 test set. Generally, the state momentum issue becomes more apparent in dialogues with larger turns, since they always contain more active slot-value pairs, and any one of the wrong pairs kept unchanged will affect the further prediction accuracy. With the increase of turns, the accuracy of Baseline harshly degrades, while MoNET gets a relatively smaller decline, resulting in a gradually increasing and evident percentage difference in accuracy. This demonstrates the superiority of MoNET in alleviating the accuracy decrease caused by the state momentum issue, especially in those dialogues with larger than 6-7 turns.

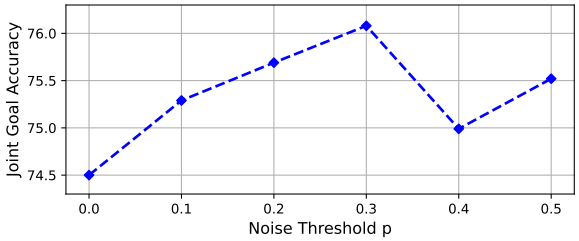

Figure 4: Performance on the MultiWOZ 2.4 validation set w.r.t the noise threshold of adding noise.

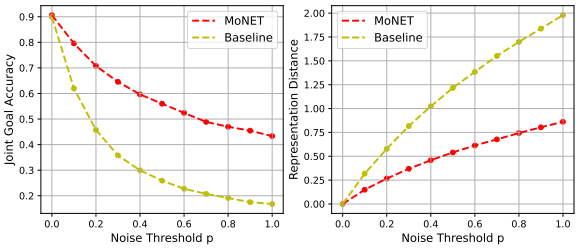

Figure 5: Performances on the MultiWOZ 2.4 test set w.r.t. the noise threshold (corresponding to the noised slot-value pair ratio in the dialogue state input). The left one is the joint goal accuracy, and the right one is the active slot-context features similarity.

5.4 Noise Threshold Selection for Training

To explore the impact of different probabilities of adding noise into the context input for training, we vary the noise threshold p from 0 to 0.5 to train our MoNET. The results on the MultiWOZ 2.4 validation set are shown in Figure 4, where MoNET achieves the best performance when the noise threshold p is set to 0.3. Intuitively, a small p makes the noised context input contain fewer noised slot-value pairs (hard to learn meaningful semantics from the noised data); conversely, a large p makes the noised context input far from the original context input in the representation space (hard to group them closer). Both two cases make the model hard to learn effective features from the noised context input, leading to lower prediction accuracy. Hence, the empirical probability of adding noise is important to derive the best performance of the DST model.

5.5 Anti-noise Probing with Noise Testing

In this section, we conduct *noise testing* to explore the impact of anti-noise ability on DST models. We first evaluate DST performances of MoNET and Baseline by introducing different ratios of noise (with p from 0 to 1) into the oracle previous dialogue state as the model input. Figure 5 shows the performances of MoNET and Baseline on MultiWOZ 2.4. Both of them get high accuracy when the noise ratio is 0, as we use the oracle previous dialogue state as the model input; with the increase of the noise ratio, the joint goal accuracy of Baseline gets a sharp decline, while MoNET degrades much more smoothly. Furthermore, for each dialogue turn, we also show the $L2$-distance between the original and noised context representations, i.e., the mean pooling of all token representations H_t and H_t^+. As can be observed, along with the increase of noise ratio, the distance between the two representations of MoNET is much lower than that of Baseline. These results indicate that MoNET achieves a higher anti-noise ability by generating relatively similar representations for the original and noised contexts, which helps the DST model maintain an acceptable performance even with a high ratio of noise in its input.

Sys: I have many trains available. What day would you like to leave? Usr: We will be leaving on Sunday afternoon. GT: train-day-Sunday Baseline: train-day-Sunday MoNET: train-day-Sunday
Sys: There are still many trains to pick from, can you narrow down a departure and arrival time frame? Usr: Yes, it should leave after 20:15 and leave on Monday, not Sunday. Please give me a price, travel time, and arrival time for any train. GT: train-day-Monday Baseline: train-day-Sunday MoNET: train-day-Monday (*update*)
Sys: Great! You are booked at Autumn House for 1 night. Your reference number is n4tvfkgs. Would you like more information? Usr: I would also like a taxi between the places if possible. GT: taxi-destination-Autumn House Baseline: taxi-destination-Gonville and Caius College MoNET: taxi-destination-Gonville and Caius College
Sys: I can help with that. When would you like to either leave or arrive? And do you want the taxi from the college to the hotel or the other way around? Usr: I need to go from the college to the hotel, and I want to leave the college by 20:45, please. GT: taxi-destination-Autumn House Baseline: taxi-destination-Gonville and Caius College MoNET: taxi-destination-Autumn House (*correct*)

Table 3: Predictions of two dialogue examples on MultiWOZ 2.4 separated by the double solid line, corresponding to two state momentum cases. Wrong and correct predicted values are marked in red and blue.

5.6 Case Study and Attention Visualization

Table 3 gives two prediction examples using MoNET and Baseline on the MultiWOZ 2.4 test set, corresponding to the two types of state momentum cases. In the first one, they correctly predict the slot-value pair "train-day-Sunday", while only MoNET updates it in the next turn along with the ground truth changing into "train-day-Monday". In the second one, they make a wrong prediction "taxi-destination-Gonville and Caius College". While

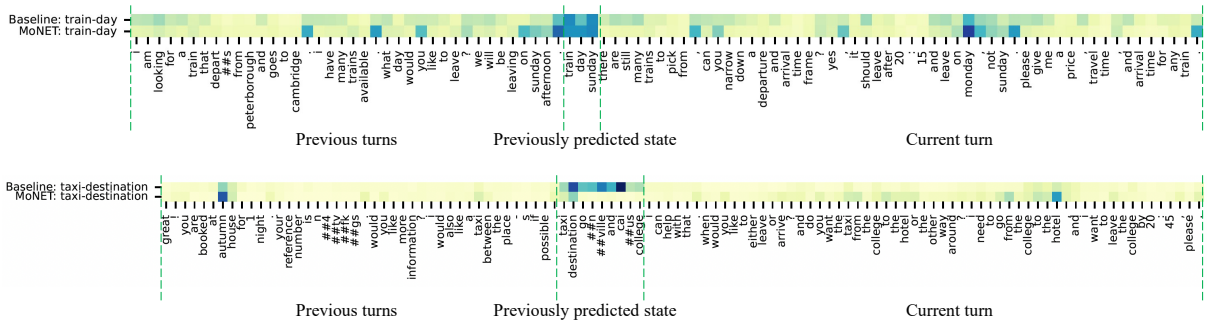

Figure 6: Attention visualizations of the two dialogue examples mentioned in Table 3.

Model	Accuracy
MinTL (Lin et al., 2020)	52.07
MTTOD (Lee, 2021)	53.56
PPTOD (Su et al., 2022)	53.37
T5-base	53.26
MoNET (T5-base)	54.67 (↑ 1.41)

Table 4: Joint goal accuracy on MultiWOZ 2.0 test set of baselines using the same T5-base pre-trained model.

Baseline keeps it unchanged in the next turn, MoNET corrects it, resulting in a joint goal accuracy of 100% for the second turn. Besides, we further explore these two examples by calculating and visualizing the overall attention scores, which are shown in Figure 6. For each slot, its overall attention score over each token is the weighted sum of the self-attended scores by all tokens in X_t. The weights come from the slot-context attention, and the self-attended scores are the average of attention scores over multiple layers in BERT. As can be observed, Baseline pays more attention to the values in the previously predicted state, and fails to solve the state momentum issues; MoNET pays relatively higher attention to the correct tokens ("monday" in the first case and "autumn house" in the second case), and consequently, successfully *updates* Sunday into Monday and *corrects* Gonville and Caius College into Autumn House. These examples and attention visualizations indicate the effectiveness of our MoNET in alleviating the two types of state momentum issues.

5.7 Extension on Generation-based Models

In addition to the original classification-based MoNET model, we also evaluate our approach using a simple generation framework using T5-base as the backbone pre-trained model (Raffel et al., 2020). The ontology is built from the database and training set annotations, which is only used for noise value construction. The model framework is similar to the BERT-based MoNET in Figure 2(a), where the BERT encoders and slot-value matching modules are replaced with T5 encoders and decoders. The T5 encoders encode the dialogue context inputs, slots, and values. After deriving the slot-context attentive representations, the T5 decoders generate each slot-value pair. Table 4 shows the joint goal accuracy performance of the T5-based MoNET on the MultiWOZ 2.0 test set, compared with other end-to-end/generation-based models using the same T5-base pre-trainied model. As can be observed, our modified MoNET outperforms the T5-base backbone and others with the same T5-base model, indicating its effectiveness and adaptability for the implementation of generation-based methods.

6 Conclusion

In this paper, we define and systematically analyze the state momentum issues in the DST task, and propose MoNET, a training strategy equipped with noised DST training and the contrastive context matching framework. Extensive experiments on MultiWOZ 2.0, 2.1, and 2.4 datasets verify its effectiveness compared with existing DST methods. Supplementary studies and analysis demonstrate that MoNET has a stronger anti-noise ability which helps alleviate the state momentum issues.

Limitations

Our proposed MoNET is a classification-based method requiring the pre-defined ontology containing all slot-value pairs. Moreover, during prediction, for each slot, its distance with all possible values is calculated, i.e., the prediction has to process 30 times, which is the number of slots in the MultiWOZ dataset. Compared

with the generation methods that only process once and do not need ontology, our method is short in training efficiency and scalability. However, most task-oriented dialogue datasets contain their knowledge base containing slot value information, so it's acceptable to construct the ontology for random sampling. Besides, the results in Section 5.7 demonstrate that our method can be implemented into generation-based backbone models.

Acknowledgements

The work was supported in part by NSFC with Grant No. 62293482, the Basic Research Project No. HZQB-KCZYZ-2021067 of Hetao Shenzhen-HK S&T Cooperation Zone, the National Key R&D Program of China with grant No. 2018YFB1800800, the Shenzhen Outstanding Talents Training Fund 202002, the Guangdong Research Projects No. 2017ZT07X152, No. 2019CX01X104, and No. 2021A1515011825, the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001), the Shenzhen Key Laboratory of Big Data and Artificial Intelligence (Grant No. ZDSYS201707251409055), and the National Key R&D Program of China under Grant No. 2020AAA0108600.

References

Paul Baltescu and Phil Blunsom. 2015. Pragmatic neural language modelling in machine translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 820–829, Denver, Colorado. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo 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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Hengyi Cai, Hongshen Chen, Yonghao Song, Zhuoye Ding, Yongjun Bao, Weipeng Yan, and Xiaofang Zhao. 2020. Group-wise contrastive learning for neural dialogue generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 793–802, Online. Association for Computational Linguistics.

Junfan Chen, Richong Zhang, Yongyi Mao, and Jie Xu. 2020a. Parallel interactive networks for multi-domain dialogue state generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1921–1931, Online. Association for Computational Linguistics.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020b. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7521–7528. AAAI Press.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020c. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Yinpei Dai, Hangyu Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, and Xiaodan Zhu. 2021. Preview, attend and review: Schema-aware curriculum learning for multi-domain dialogue state tracking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 879–885, Online. Association for Computational Linguistics.

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

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

Yue Feng, Aldo Lipani, Fanghua Ye, Qiang Zhang, and Emine Yilmaz. 2022. Dynamic schema graph fusion network for multi-domain dialogue state tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume*

1: Long Papers), pages 115–126, Dublin, Ireland. Association for Computational Linguistics.

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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

Tassilo Klein and Moin Nabi. 2020. Contrastive self-supervised learning for commonsense reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7517–7523, Online. Association for Computational Linguistics.

Byung-Jun Lee and Kee-Eung Kim. 2016. Dialog history construction with long-short term memory for robust generative dialog state tracking. *Dialogue & Discourse*, 7(3):47–64.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue state tracking with a language model using schema-driven prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4937–4949, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Yohan Lee. 2021. Improving end-to-end task-oriented dialog system with a simple auxiliary task. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1296–1303, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Fatema Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2021. Coco: Controllable counterfactuals for evaluating dialogue state trackers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Weizhe Lin, Bo-Hsiang Tseng, and Bill Byrne. 2021. Knowledge-aware graph-enhanced GPT-2 for dialogue state tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7871–7881, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th*

International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012. icml.cc / Omnipress.

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

Yawen Ouyang, Moxin Chen, Xinyu Dai, Yinggong Zhao, Shujian Huang, and Jiajun Chen. 2020. Dialogue state tracking with explicit slot connection modeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 34–40, Online. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Abhinav Rastogi, Dílek Hakkani-Tür, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 561–568.

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

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676, Dublin, Ireland. Association for Computational Linguistics.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022. BORT: Back and denoising reconstruction for end-to-end task-oriented dialog. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2156–2170, Seattle, United States. Association for Computational Linguistics.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2023. Mars: Semantic-aware contrastive learning for end-to-end task-oriented dialog. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada. Association for Computational Linguistics.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Dong Wang, Ning Ding, Piji Li, and Haitao Zheng. 2021. CLINE: Contrastive learning with semantic negative examples for natural language understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2332–2342, Online. Association for Computational Linguistics.

Yexiang Wang, Yi Guo, and Siqi Zhu. 2020. Slot attention with value normalization for multi-domain dialogue state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3019–3028, Online. Association for Computational Linguistics.

Yifan Wang, Jing Zhao, Junwei Bao, Chaoqun Duan, Youzheng Wu, and Xiaodong He. 2022. LUNA: Learning slot-turn alignment for dialogue state tracking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3319–3328, Seattle, United States. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*,

pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Jason D. Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

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

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

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. UBAR: towards fully end-to-end task-oriented dialog system with GPT-2. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14230–14238. AAAI Press.

Fanghua Ye, Yue Feng, and Emine Yilmaz. 2022a. ASSIST: Towards label noise-robust dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2719–2731, Dublin, Ireland. Association for Computational Linguistics.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2022b. MultiWOZ 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 351–360, Edinburgh, UK. Association for Computational Linguistics.

Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021. Slot self-attentive dialogue state tracking. In *Proceedings of the Web Conference 2021*, WWW '21, page 1598–1608, New York, NY, USA. Association for Computing Machinery.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

Tao Yu, Rui Zhang, Alex Polozov, Christopher Meek, and Ahmed Hassan Awadallah. 2021. Score: Pre-training for context representation in conversational semantic parsing. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yan Zeng and Jian-Yun Nie. 2020. Jointly optimizing state operation prediction and value generation for dialogue state tracking. *ArXiv preprint*, abs/2010.14061.

Haoning Zhang, Junwei Bao, Haipeng Sun, Huaishao Luo, Wenye Li, and Shuguang Cui. 2022. CSS: Combining self-training and self-supervised learning for few-shot dialogue state tracking. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 302–310, Online only. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, Philip Yu, Richard Socher, and Caiming Xiong. 2020. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 154–167, Barcelona, Spain (Online). Association for Computational Linguistics.

Su Zhu, Jieyu Li, Lu Chen, and Kai Yu. 2020. Efficient context and schema fusion networks for multi-domain dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 766–781, Online. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The final section after Conclusion

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 4

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
No response.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
No response.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
No response.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.