Multi-Domain Dialogue State Tracking with Top-K Slot Self Attention

Longfei Yang[1], **Jiyi Li**[2], **Sheng Li**[3], **Takahiro Shinozaki**[1]
[1]Tokyo Institute of Technology
[2] University of Yamanashi
[3] National Institute of Information and Communications Technology
longfei.yang.cs@gmail.com, jyli@yamanashi.ac.jp, sheng.li@nict.go.jp,
shinot@ict.e.titech.ac.jp

Abstract

As an important component of task-oriented dialogue systems, dialogue state tracking is designed to track the dialogue state through the conversations between users and systems. Multi-domain dialogue state tracking is a challenging task, in which the correlation among different domains and slots needs to consider. Recently, slot self-attention is proposed to provide a data-driven manner to handle it. However, a full-support slot self-attention may involve redundant information interchange. In this paper, we propose a top-k attention-based slot self-attention for multi-domain dialogue state tracking. In the slot self-attention layers, we force each slot to involve information from the other k prominent slots and mask the rest out. The experimental results on two mainstream multi-domain task-oriented dialogue datasets, MultiWOZ 2.0 and MultiWOZ 2.4, present that our proposed approach is effective to improve the performance of multi-domain dialogue state tracking. We also find that the best result is obtained when each slot interchanges information with only a few slots.

1 Introduction

As a crucial component of task-oriented dialogue systems, dialogue state tracking (DST) is designed to track the dialogue states through the conversations between users and systems (Young et al., 2010, 2013), which is generally expressed as a list of {(domain, slot, value)}. In recent years, dialogue state tracking has drawn more and more attention, and numerous methods are proposed (Mrkšić et al., 2017; Zhong et al., 2018; Nouri and Hosseini-Asl, 2018; Ramadan et al., 2018).

Despite many progresses have been achieved, these approaches track dialogue states for each slot separately without considering the correlation among slots (Ouyang et al., 2020; Wu et al., 2019; Lee et al., 2019; Hu et al., 2020; Ye et al., 2021b). Spoken language is not formal, in which ellipsis and cross-reference phenomena make multi-domain dialogue state tracking problematic as shown in Table 1. To provide the user with several options, the values of slot "attraction-area" in the domain "attraction" at the second turn, the system should look for the information in another domain "restaurant" because the user implicitly indicates that the attraction he is looking for should be near the restaurant without explicitly speaking it out. And the value of slot "taxi-destination" should be that the system mentioned at the first turn.

User: Hi, I'm looking for a cheap restaurant in the centre of the city. **Sys:** Nutnut is a steal and popular there. **State:** restaurant-area=centre; restaurant-pricerange=cheap; restaurant-name=nutnut
User: Is there any place of pleasure near it? **Sys:** What type of attraction do you like? State: **restaurant-area=centre**; restaurant-pricerange=cheap; restaurant-name=nutnut; **attraction-area=centre**;
.....
User: Can you book a taxi for me to get to the restaurant? **Sys:** Of course, could you please provide your departure place? **State:** restaurant-area=centre; restaurant-pricerange=cheap; **restaurant-name=nutnut**; attraction-area=centre; **taxi-destination=nutnut**

Table 1: An example of a dialogue with three domains.

Several researchers have paid attention to modeling the correlations to some certain degrees (Ouyang et al., 2020; Hu et al., 2020; Heck et al., 2020). In these works, the correlation between the slot names is taken into consideration (Ouyang et al., 2020) or a strong prior knowledge is involved, i.e., the similarity coefficient is set to one manually if two slots are regarded to be relevant by human (Hu et al., 2020). But it may overlook the

Proceedings of the SIGdial 2022 Conference, pages 231–236
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

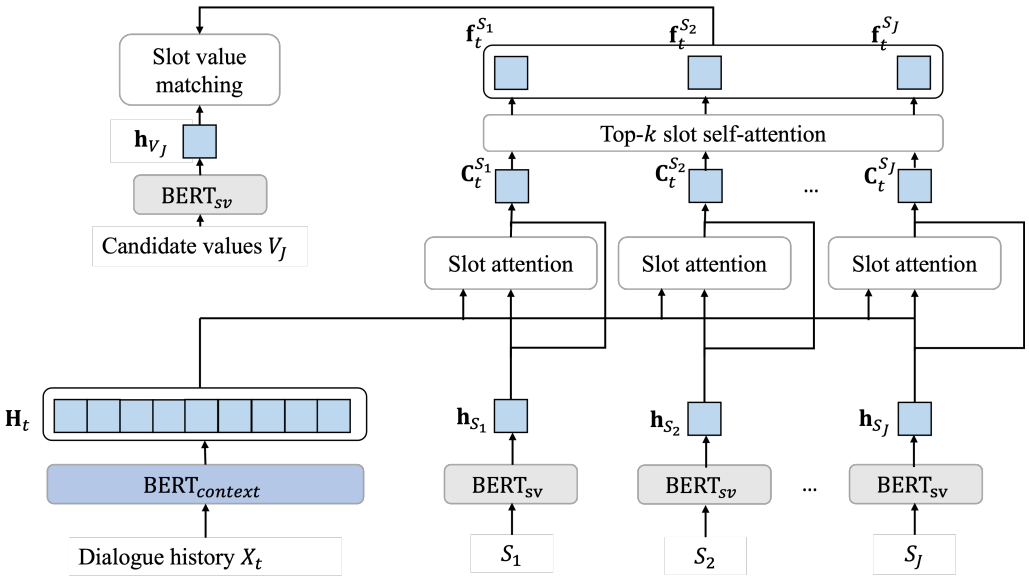

Figure 1: An overview of the proposed approach. For the BERT$_{context}$ model of the context encoder (solid blue rounded rectangle), its parameters are fine-tuned during training to encoder dialogue history; for the BERT$_{sv}$ model of the slot-value encoder (gray rounded rectangle), its parameters remain frozen to encoder slots and candidate values.

dependencies of some slots with the approach only considering the slot names. To address it, Ye et al. (2021b) proposed a slot self-attentive attention extracting slot-specific information for each slot from the dialogue context by utilizing a stacked slot self-attention module to learn the correlations among slots in a fully data-driven way without any human efforts or prior knowledge. However, it may involve some redundant information for some specific slots from other slots and result in incorrect prediction.

In this paper, we propose a dialogue state tracking with top-k slot self-attention. Here we have a premise of this work: *For each slot, not all of the others play a positive role in the value prediction for it. The more redundant information is involved, the worse would be the performance.* More specifically, in our work, in the layer where the slots interchange their information, we force each slot to pay attention to the other k slots with the highest scores and mask the rest out rather than considering all of them. We conduct experiments on MultiWOZ 2.0 and MultiWOZ 2.4 datasets and present that our proposed model works better than the methods handling the correlations with fully slot self attention.

The contributions of this paper are as follows: (1) We propose a top-k attention-based slot self-attention method for multi-domain dialogue state tracking; (2) The experimental results verify the effectiveness of our approach, and we find that the best result is obtained when each slot interchanges information with only a few slots.

2 Approach

Figure 1 shows the overview of the proposed model. It consists of a dialogue encoder, slot attention, top-k slot self-attention, and slot value matching.

2.1 Encoding

Let's define the dialogue history $D_T = \{R_1, U_1, ..., R_T, U_T\}$ as a set of system responses R and user utterances U in T turns of dialogue, where $R = \{R_t\}_{t=1}^{T}$ and $U = \{U_t\}_{t=1}^{T}, 1 \leq t \leq T$. We define $E_T = \{B_1, ..., B_T\}$ as the dialogue states of T turns, and each E_t is a set of slot value pairs $\{(S_1, V_1), ..., (S_J, V_J)\}$ of J slots. The context encoder accepts the dialogue history till turn t, which can be denoted as $X_t = \{D_1, ..., D_t, E'_{t-1}\}$, as the input and generates context vector representations $\mathbf{H}_t$.

$$\mathbf{H}_t = \text{BERT}_{context}(X_t) \quad (1)$$

Another BERT$_{sv}$ is employed to encode the slots and candidate values. The difference is that the parameters of BERT$_{sv}$ remain frozen during training. For those slots and values containing multiple tokens, the vector corresponding to the special token [CLS] is employed to represent them. For each slot

S_j and value V_j,

$$\mathbf{h}_{S_j} = \text{BERT}_{sv}(S_j) \quad (2)$$

$$\mathbf{h}_{V_j} = \text{BERT}_{sv}(V_j) \quad (3)$$

2.2 Slot attention

For predicting the states of a specific slot, it is necessary to extract slot-specific information from the dialogue history (**?**). A multi-head attention-based slot attention is employed to capture this information.

$$\mathbf{Q}_t^{S_j} = \mathbf{h}_{S_j}\mathbf{W}_Q + \mathbf{b}_Q \quad (4)$$

$$\mathbf{K}_t^{S_j} = \mathbf{H}_t\mathbf{W}_K + \mathbf{b}_K \quad (5)$$

$$\mathbf{V}_t^{S_j} = \mathbf{H}_t\mathbf{W}_V + \mathbf{b}_V \quad (6)$$

$$\boldsymbol{\alpha}_t^{S_j} = Softmax(\frac{\mathbf{Q}_t^{S_j}{\mathbf{K}_t^{S_j}}^T}{\sqrt{d_k}})\mathbf{V}_t^{S_j} \quad (7)$$

$$\mathbf{C}_t^{S_j} = \mathbf{W}_2 ReLU(\mathbf{W}_1[\mathbf{h}_{S_j}, \boldsymbol{\alpha}_t^{S_j}] + \mathbf{b}_1) + \mathbf{b}_2 \quad (8)$$

Where $\mathbf{W}_Q, \mathbf{b}_Q, \mathbf{W}_K, \mathbf{b}_K$, $\mathbf{W}_V$, and $\mathbf{b}_V$ are the parameters of the linear layers for projecting query, key and value respectively. $d_k = d_h/N$ in which d_h is the hidden size of the model, and N is the number of heads in multi-head attention.

2.3 Top-k slot self-attention

Inspired by Ye et al. (2021b), the information across different slots can be communicated by applying self-attention mechanism. In this work, we introduce a top-k slot self-attention to capture the correlation among different slots. We assume that, for each slot, not all of the other slots play a positive role in the value prediction. Forcing it with a few k slots with the highest attention scores performs better than considering all of them. To implement it, we mask out all but its k largest dot products with the keys in the slot-attention layers. For the l-th self-attention sub-layer, $\mathbf{F}_t^l = [\mathbf{C}_t^{S_1}, ..., \mathbf{C}_t^{S_J}]$, the formulations are as follows.

$$\widetilde{\mathbf{F}}_t^l = LayerNorm(\mathbf{F}_t^l) \quad (9)$$

$$\mathbf{G}_t^l = TopkAtt(\widetilde{\mathbf{F}}_t^l, \widetilde{\mathbf{F}}_t^l, \widetilde{\mathbf{F}}_t^l) + \widetilde{\mathbf{F}}_t^l \quad (10)$$

$$TopkAtt(Q, K, V) = Softmax(Topk(QK^\top))V \quad (11)$$

For the l-th feed-forward sub-layer, the formulations are as follows.

$$\widetilde{\mathbf{G}}_t^l = LayerNorm(\mathbf{G}_t^l) \quad (12)$$

$$\mathbf{F}_t^{l+1} = FFN(\widetilde{\mathbf{G}}_t^l) + \widetilde{\mathbf{G}}_t^l \quad (13)$$

The output of the final layer is regarded as the final slot specific vector $\mathbf{F}_t^{L+1} = [\mathbf{f}_t^{S_1}, ..., \mathbf{f}_t^{S_J}]$, where $\mathbf{f}_t^{S_j}$ represents the output corresponding to a slot.

2.4 Slot value matching

A Euclidean distance-based value prediction is performed for each slot. Firstly, the slot-specific vector is fed into a normalization layer. Then the distances between slot-specific vector and value are measured. Finally, the nearest value is chosen to predict the state value.

$$\boldsymbol{r}_t^{S_j} = LayerNorm(Linear(\mathbf{f}_t^{S_j})), \quad (14)$$

$$p(V_t^j|X_t, S_j) = \frac{\exp(-d(\mathbf{h}^{V_j}, \boldsymbol{r}_t^{S_j}))}{\sum_{V_j' \in \nu_j} \exp(-d(\mathbf{h}^{V_j'}, \boldsymbol{r}_t^{S_j}))} \quad (15)$$

where $d(\cdot)$ is Euclidean distance function, and ν_j denotes the value space of the slot S_j. The model is trained to maximize the joint probability of all slots. The loss function at each turn t is denoted as the sum of the negative log-likelihood.

$$\mathcal{L}_t = \sum_{j=1}^{J} -\log(p(V_t^j|X_t, S_j)) \quad (16)$$

3 Experiments

3.1 Datasets

MultiWOZ 2.0 (Budzianowski et al., 2018) and MultiWOZ 2.4 (Ye et al., 2021a) are employed as the datasets in our experiments. MultiWOZ 2.0 is one of the largest open-source human-human conversational datasets of multiple domains. It contains over 10,000 dialogues in which each dialogue averages 13.68 turns. MultiWOZ 2.4 is the latest refined version. It mainly fixes the annotation errors in the validation and test set. To make a fair comparison with the models evaluated on these two datasets, we follow the procedure in several previous works (Wu et al., 2019; Lee et al., 2019; Wang et al., 2020; Ye et al., 2021b) to keep consistent.

3.2 Training details

The used dialogue context encoder $\text{BERT}_{context}$ is a pre-trained BERT-base-uncased model of 12 layers with 768 hidden units and 12 self-attention heads. We employ another BERT-base-uncased model as the slot and value encoder BERT_{sv}. The

Table 2: The joint goal accuracy (JGA) of different models on the test set of MultiWOZ 2.0 and 2.4 dataset.

	Model	MW2.0	MW2.4
Open Vocabulary	TRADE (Wu et al., 2019)	48.62	54.89
	SOM (Kim et al., 2020)	51.72	66.78
	TripPy (Heck et al., 2020)	-	59.62
	SimpleTOD (Hosseini-Asl et al., 2020)	-	66.78
Ontology	SUMBT (Lee et al., 2019)	46.65	61.86
	DS-DST (Zhang et al., 2020)	52.24	-
	DS-Picklist (Zhang et al., 2020)	54.39	-
	SAVN (Wang et al., 2020)	54.52	60.55
	SST (Chen et al., 2020)	51.17	-
	STAR (Ye et al., 2021b)	54.53	73.94
	Top-k SSA(k=1)	**54.82**	77.10
	Top-k SSA(k=3)	54.47	**77.25**

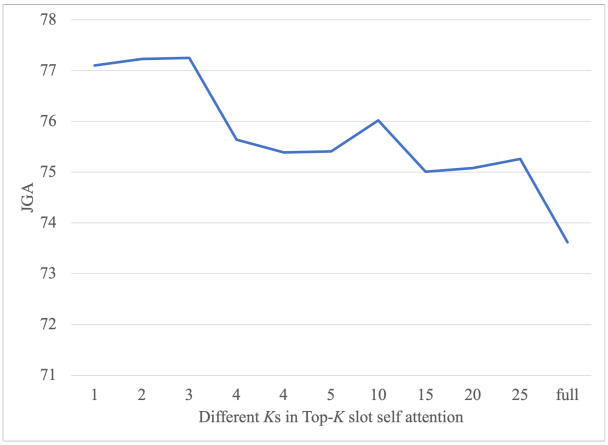

Figure 2: The results (JGA) of the proposed model based on top-k slot self-attention with different ks on MultiWOZ 2.4 dataset.

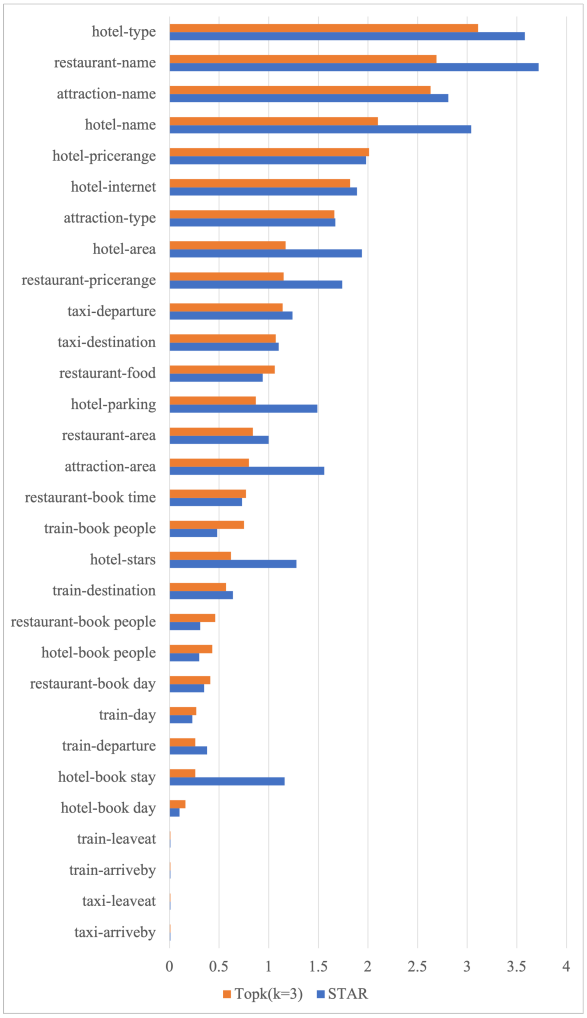

Figure 3: The error rate of each slot in the STAR and our proposed model on MultiWOZ 2.4 dataset.

number of attention heads for slot attention and slot self-attention is 4. The number of slot self-attention layers is 6. Adam optimizer is adopted with a training batch size of 8. The hidden size is set to 768 for the model. The slot attention part has 6 layers in which the number of attention heads is 6 as well. Adam is used as the optimizer with a learning rate of 4e-5 for encoder and 1e-4 for other parts. The hyper-parameters are chosen from the best-performing model over the validation set. We use the training batch size 16 and dropout rate 0.1 on utterances in a dialogue history.

4 Results and analysis

4.1 Main results

Table 2 shows the main results. We compare our approach with several typical and SOTA methods on this task. Top-k SSA denotes our proposed model with top-k slot self-attention. Joint goal accuracy (JGA) is employed to evaluate the overall performance. The joint goal accuracy is a strict measurement comparing the predicted values of each slot with ground truth for each dialogue turn, and the prediction is considered correct if and only if all the predicted values match the ground truth values without any error at each turn. As shown in Table 2, our proposed model achieves the best performance on these two datasets. We utilize the Wilcoxon signed-rank test, and the proposed approach is statistically significantly better ($p < 0.05$) than baselines. Specifically, our model with top-k (k=1) SSA for the MultiWOZ 2.0 dataset obtains a JGA of 54.82%. For the refined MultiWOZ 2.4 dataset, our model with top-k (k=3) SSA achieves a JGA of 77.25%, which outperforms other models by a large margin.

4.2 The effect of different ks

We investigate the performance using different ks to have a further understanding. As shown in Figure 2, the best performance is obtained when k is small, which means each slot performs infor-

User: Hi, I am looking for a place to eat some indian food. **Sys:** Do you have a price range in mind?
.....
User: I would like a place in the south, please. **Sys:** Taj Tandoori is the place you want to go. It meets all of your needs. STAR: restaurant-name=none Ours: restaurant-name=taj tandoori
.....
User: I want a taxi from the restaurant that I am at. **Sys:** Ok, so you would like a taxi from the restaurant to the park? Could you please let me know your desired departure and arrival times? STAR: taxi-departure=tandoori palace Ours: taxi-departure=taj tandoori
User: I am sorry, I would like a taxi from Wandlebury country to Taj Tandoori. I would like the taxi to pick me up at 10:15. **Sys:** Okay, I have booked a taxi for you it will be white tesla ... STAR: taxi-departure=tandoori palace; taxi-destination=Wandlebury country Ours: taxi-departure=Wandlebury country; taxi-destination=taj tandoori

Table 3: An example of a dialogue MUL2491 in MultiWOZ 2.4 dataset.

mation interchange with only a few slots. Then it drops a lot with the increase of k. It verifies our assumption that it is positive to force each slot to interchange information with limited slots than all of them to prevent abundant information interchange, in which the more redundant information is involved, the worse would be the performance.

4.3 Error analysis

Figure 3 presents the error rate of each slot. First it can be noticed that the overall error rate is reduced with our model. We also find that, comparing with the previous SOTA model STAR, the performance of the slots that may interchange information with others, e.g., *hotel-area*, *restaurant-area*, is improved by a large margin with our model. The performance of *"taxi"*-related and *"train"*-related is also improved slightly. Even though our model reduce the error rates of several *"name"*-related slots, like *"restaurant-name"* and *"hotel-name"*, they still have very high error rate.

4.4 Case study

Table 3 demonstrates an example in the test set of MultiWOZ 2.4 dataset. We can note that firstly STAR makes a mistake in the prediction for the slot *"restaurant-name"* while our model correctly find it. At the last turn, the user indicates his/her *"taxi-departure"* and *"taxi-destination"*. Although STAR capture the "Wandlebury country" but it fails to find the correlation of these slots, and the value of *"restaurant-name"* is copied from the error predicted value for this slot at the previous turns.

5 Conclusion

In this work, to address the correlation among different slots, we propose multi-domain dialogue state tracking with top-k slot self-attention, in which, each slots is forced to interchange information with the k slots with highest scores than all of them. We conduct experiments on MultiWOZ 2.0 and MultiWOZ 2.4 datasets and present that our model works better than existing methods that consider the correlations. The best results can be obtained when each slot interchanges information with only a few other slots.

Acknowledgements

This study was partly supported by JSPS KAKENHI Grand Number JP22K12069.

References

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.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7521–7528.

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.

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, volume 33, pages 20179–20191.

Jiaying Hu, Yan Yang, Chencai Chen, Liang He, and Zhou Yu. 2020. SAS: Dialogue state tracking via slot attention and slot information sharing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6366–6375.

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.

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.

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*, pages 1777–1788.

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

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.

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

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*, pages 3019–3028.

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.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2021a. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. *arXiv preprint arXiv:2104.00773*.

Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021b. Slot self-attentive dialogue state tracking. In *Proceedings of the Web Conference 2021*, pages 1598–1608.

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.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

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.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1458–1467.