Multi-Domain Dialogue State Tracking with Disentangled Domain-Slot 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 the core of task-oriented dialogue systems, dialogue state tracking (DST) is designed to track the dialogue state through the conversation between users and systems. Multi-domain DST has been an important challenge in which the dialogue states across multiple domains need to consider. In recent mainstream approaches, each domain and slot are aggregated and regarded as a single query feeding into attention with the dialogue history to obtain domain-slot specific representations. In this work, we propose disentangled domain-slot attention for multi-domain dialogue state tracking. The proposed approach disentangles the domain-slot specific information extraction in a flexible and context-dependent manner by separating the query about domains and slots in the attention component. Through a series of experiments on MultiWOZ 2.0 and MultiWOZ 2.4 datasets, we demonstrate that our proposed approach outperforms the standard multi-head attention with aggregated domain-slot query.

1 Introduction

Task-oriented dialogue system is designed to assist users to accomplish sorts of certain tasks. For example, by using dialogue-based automated customer service, users can online query information and make reservations. Multi-domain dialogue state tracking has been an important challenge introduced by Budzianowski et al. (2018), in which numerous mixed-domain conversations are involved. In this case, DST has to track the dialogue states at each turn through the conversation, which contains a huge space involving the combinations of the ontology of different domains, slots, and values. It is a challenging task since spoken language is not formal, in which ellipsis and cross-reference are barrier to handling the correlations among different domains and slots.

Several studies have explored sorts of approaches to handle the correlations among domains and slots. In recent mainstream approaches, each domain and slot are aggregated into a single vector regarded as a query. The query and the dialogue history are fed into attention to generate domain-slot specific representations (Wu et al., 2019). Then the information interchange across different domains and slots are performed with them to model the correlation among different domain and slots (Hu et al., 2020; Wang and Lemon, 2013; Ye et al., 2021). However, these approaches introduce too much human prior knowledge and they only consider the correlations among domains and slots names or overestimate these correlations (Yang et al., 2022).

To tackle this problem, we propose a disentangled domain-slot attention (DDSA), which disentangles information extraction about domains and slots in a flexible and context-dependent manner. In detail, we disentangle the query about domains and slots in the domain-slot attention component. Firstly, domain specific representations are obtained using the domain query and the dialogue history. Then the model utilizes these representations and slot query to retrieve slot specific information (in this context, slot means the slot only) and generate domain-slot specific representations. Finally, state prediction is performed with these domain-slot specific representations.

We conduct experiments to verify our approach on MultiWOZ 2.0 and MultiWOZ 2.4 datasets. The experimental results show that the proposed approach can effectively improve the performance of multi-domain dialogue state tracking. The contributions of this work can be addressed as follows. (1) We propose a disentangled domain-slot attention mechanism to handle the correlations among domains and slots, in which the process of domain-slot specific information extraction is disentangled in a flexible and context-dependent manner. (2) We demonstrate that the performance of DST benefits from our proposed approach and make a detailed empirical study that shows that our model performs

Findings of the Association for Computational Linguistics: ACL 2023, pages 4928–4938
July 9-14, 2023

better than the baseline models based on standard attention with aggregated domain-slot query[1].

2 Related Works

Dialogue state tracking (DST) is the core of task-oriented dialogue systems. In the early years, DST highly relies on hand-crafted semantic features to predict the dialogue states (Williams and Young, 2007; Thomson and Young, 2010; Wang and Lemon, 2013), which is hard to handle lexical and morphological variations in spoken language (Lee et al., 2019). Benefiting from the rapid development of deep learning methods, neural network-based DST models have been explored. Mrkšić et al. (2017) proposes a novel neural belief tracking (NBT) framework with learning n-gram representations of utterances. Inspired by it, a lot of neural network models are investigated (Nouri and Hosseini-Asl, 2018; Ren et al., 2018; Zhong et al., 2018; Hu et al., 2020; Ouyang et al., 2020; Wu et al., 2019) and achieve further improvement.

Pre-trained models have brought natural language processing to a new era in recent years. Many substantial works have shown that the pre-trained models can learn universal language representations, which are beneficial for downstream tasks (Mikolov et al., 2013; Pennington et al., 2014; McCann et al., 2017; Sarzynska-Wawer et al., 2021; Devlin et al., 2019; Mittal et al., 2021). More recently, the very deep pre-trained language models, such as Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2019) and Generative Pre-Training (GPT) (Radford et al., 2018), trained with an increasing number of self-supervised tasks have been proposed to make the models capturing more knowledge from a large scale of corpora, which have shown their abilities to produce promising results. In view of it, many pieces of studies about DST have explored to establish the models on the basis of these pre-trained language models (Hosseini-Asl et al., 2020; Kim et al., 2020; Lee et al., 2019; Zhang et al., 2020; Chen et al., 2020; Chao and Lane, 2019; Ye et al., 2021; Heck et al., 2020; Lin et al., 2020).

Related to handling the correlations among domains and slots in multi-domain DST, several approaches have been investigated. In recent mainstream approaches, domain-slot specific representations are first achieved using attention mechanism with aggregated domain-slot query, and then the correlations are modeled with them. (Balaraman and Magnini, 2021) utilizes domain and slot information to extract both domain and slot specific representations and then combines such representations to predict the values. Chen et al. (2020) manually constructs a schema graph modeling the dependencies of different slots and introduces a graph attention matching network to mix the information from utterances and graphs to control the state updating. Hu et al. (2020) introduces a matrix representing the similarity among different slots and then perform slot information sharing among similar slots. The above two approaches are name-based since they only consider the semantics dependencies of slot names to measure the correlation among different slots, which may result in overlooking the dependencies of some slots. More recently, Ye et al. (2021) proposes a data-driven approach to handle these correlations, in which slot self-attention is introduced. However, this approach may inevitably result in overestimating some correlations (Yang et al., 2022).

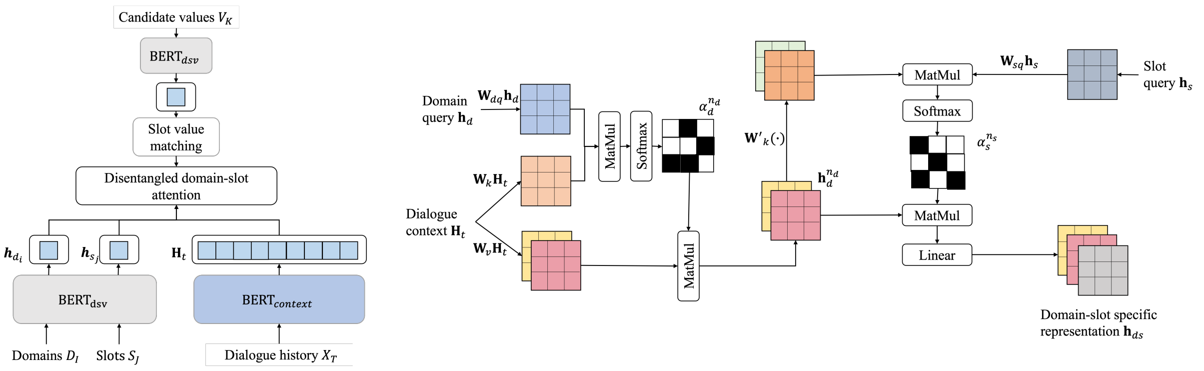
3 Dialogue State Tracking with Disentangled Domain-Slot Attention

Figure 1(a) presents the overview of the proposed model. It consists of a dialogue encoder, a domain, slot and value encoder, disentangled domain-slot attention (DDSA), and slot value matching. The context representations of dialogue history, domains, slots and values are firstly obtained by feeding dialogue history, domains, slots and values into encoders respectively. And then these representations are passed to our proposed disentangled domain-slot attention, as shown detailedly in Figure 1b, to achieve domain-slot specific representations. Finally, the corresponding values are chosen to predict the state values with these representations and slot value matching.

3.1 Encoding

We employ BERT as the encoder to generate semantic representations. The $\text{BERT}_{context}$ whose parameters are fine-tuned during training is used for encoding the dialogue context. Let's define the dialogue context history $C_T = \{R_1, U_1, ..., R_T, U_T\}$ 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

[1]The code is available at `https://github.com/couragelfyang/DDSA`



(a) An overview of the framework.

(b) A demonstration of our proposed disentangled domain-slot attention.

Figure 1: A demonstration of the model with our proposed disentangled domain-slot Attention.

turns, and each E_t is a set of slot value pairs $\{(S_1, V_1), ..., (S_J, V_J)\}$ of J slots. Although the dialogue history $C_t = \{R_t, U_t\}$ contains integrated information for the conversation until the t-th turn, the previous study (Ye et al., 2021) has indicated that it is helpful to combine it along with a compact representation E'_{t-1}, which only includes the slots whose values are not none, as part of the input. In view of this, the context encoder accepts the dialogue history till turn t, which can be denoted as $X_t = \{C_t, E'_{t-1}\}$, as the input and generates context vector representations $\mathbf{H}_t = \text{BERT}_{context}(X_t)$.

Another pre-trained model BERT$_{dsv}$ is employed to encode the domains, slots, and candidate values, in which the parameters of BERT$_{dsv}$ remain frozen. For those slots and values containing multiple tokens, the vector corresponding to the special token [CLS] is employed to represent them. For each domain D_i slot S_j and value V_k, $\mathbf{h}_{d_i} = \text{BERT}_{dsv}(D_i)$, $\mathbf{h}_{s_j} = \text{BERT}_{dsv}(S_j)$, $\mathbf{h}_{v_k} = \text{BERT}_{dsv}(V_k)$.

3.2 Disentangled Domain-Slot Attention

Figure 1(b) demonstrates the structure of our proposed disentangled domain-slot attention. The extraction with query about domains and slots is disentangled into two stages. The domain specific representations are first obtained using the domain query and the dialogue context. The slot query is employed to retrieve slot specific information based on the output of the previous stage. Finally, domain-slot specific context representations are achieved for the subsequent state prediction.

3.2.1 Domain Query

Domain specific representations are achieved using the hidden representations of domains $\mathbf{h}_d$ and that of dialogue context $\mathbf{H}_t$[2]. The process can be described as follows:

$$\mathbf{Q}_d = \mathbf{W}_{dq}^{n_d}\mathbf{h}_d + \mathbf{b}_{Q_d} \tag{1}$$

$$\mathbf{K}_d = \mathbf{W}_{K_d}^{n_d}\mathbf{H}_t + \mathbf{b}_{K_d} \tag{2}$$

$$\mathbf{V}_d = \mathbf{W}_{V_d}^{n_d}\mathbf{H}_t + \mathbf{b}_{V_d} \tag{3}$$

$$\boldsymbol{\alpha}_d^{n_d} = softmax(\frac{\mathbf{Q}_d\mathbf{K}_d^\intercal}{\sqrt{k_{dim}}}, axis = domain) \tag{4}$$

$$\mathbf{h}_d^{n_d} = \boldsymbol{\alpha}_d^{n_d}\mathbf{V}_d \tag{5}$$

Where $\mathbf{W}_{dq}, \mathbf{b}_{Q_d}, \mathbf{W}_{K_d}, \mathbf{b}_{K_d}, \mathbf{W}_{V_d}, \mathbf{b}_{V_d}$ are the parameters of the linear layers for projecting query, key, and value respectively at the domain query stage. $k_{dim} = k_{model}/n_d$, in which k_{model} is the hidden size of the model and $n_d \in N_d$ is the heads of the multi-head dot-product attention at this stage.

3.2.2 Slot Query

After the domain query stage, slot specific representations can be obtained using the output of the domain query stage and the hidden representations of slots $\mathbf{h}_s$. Note that here "slot" means the slot only rather than the concatenation or the average on the representations of domains and slots pairs. The process is shown as follows:

$$\mathbf{Q}_s = \mathbf{W}_{sq}^{n_s}\mathbf{h}_s + \mathbf{b}_{Q_s} \tag{6}$$

$$\mathbf{K}_s = \mathbf{W'}_{K_s}^{n_s}\mathbf{h}_d^{n_d} + \mathbf{b}_{K_s} \tag{7}$$

$$\mathbf{V}_s = \mathbf{h}_d^{n_d} \tag{8}$$

$$\boldsymbol{\alpha}_s^{n_s} = softmax(\frac{\mathbf{Q}_s\mathbf{K}_s^\intercal}{\sqrt{k_{ddsa}}}, axis = slot) \tag{9}$$

$$\mathbf{h}_{ds}^{n_s} = \boldsymbol{\alpha}_s^{n_s}\mathbf{V}_s \tag{10}$$

$$\mathbf{h}_{ds} = \mathbf{W}_{os}Concat(\mathbf{h}_{ds}^1, ..., \mathbf{h}_{ds}^{N_s}) \tag{11}$$

[2]Here we omit the indices of domains and slots for simplification.

Where $\mathbf{W}_{sq}, \mathbf{b}_{Q_s}, \mathbf{W'}_{K_s}, \mathbf{b}_{K_s}, \mathbf{W}_{V_s}, \mathbf{b}_{V_s}$ are the parameters of the linear layers for projecting query, key and value respectively at the slot query stage, and $\mathbf{W}_{os}$ is the parameters of the linear layer for aggregating the heads of slot query. k_{ddsa} is a hyperparameter indicating the hidden dimension in this component, and $n_s \in N_s$ is the number of heads at this stage.

Since the number of combinations of domains and slots is generally larger than that of the actual domain-slot pairs, a linear layer is employed to project domain-slot specific representation $\mathbf{h}_{ds}$ to the representation of the actual size.

$$\mathbf{h}_{ds} = \mathbf{W}_{od} Concat(\mathbf{h}_{ds}^1, ..., \mathbf{h}_{ds}^{N_d}) \tag{12}$$

$$\mathbf{h'}_{ds} = Linear(\mathbf{h}_{ds}, axis = domain \times slot) \tag{13}$$

Where $\mathbf{W}_{od}$ is the parameters of the linear layer for aggregating the heads of domain query.

3.3 Slot Value Matching

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

$$\boldsymbol{r}_t^{DS_m} = LayerNorm(Linear(\mathbf{h'}_{ds})), \tag{14}$$

$$p(V_t^k | X_t, DS_m) = \frac{\exp(-d(\mathbf{h}^{V_k}, \boldsymbol{r}_t^{DS_m}))}{\sum_{V'_k \in \nu_k} \exp(-d(\mathbf{h}^{V'_k}, \boldsymbol{r}_t^{DS_m}))} \tag{15}$$

where $d(\cdot)$ is Euclidean distance function, and ν_k denotes the value space of the actual domain-slot DS_m. 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.

$$\mathcal{L}_t = \sum_{m=1}^{M} -\log(p(V_t^k | X_t, DS_m)) \tag{16}$$

4 Experimental Settings

We conduct the experiments using MultiWOZ 2.0 and MultiWOZ 2.4 datasets in this work. MultiWOZ 2.0 (Budzianowski et al., 2018) 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 (Ye et al., 2022). 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 pre-processing and evaluation procedure in several previous works (Wu et al., 2019; Lee et al., 2019; Wang et al., 2020; Ye et al., 2021) to keep consistent. We present the settings of the model in Appendix A.

5 Results and Discussions

5.1 Main Results

Joint goal accuracy (JGA) and slot accuracy (SA) are 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. The slot accuracy compares each value to the corresponding ground truth individually without seeing other turns. For the results of baselines, we use the results reported in the corresponding references.

Table 1 presents the results of the different models on the test set of MultiWOZ 2.0 and 2.4 datasets. As shown in it, overall, our proposed model achieves the best performance on these two datasets. We utilize the Wilcoxon signed-rank test, the proposed method is statistically significantly better ($p < 0.05$) than baselines. Comparing to the previous SOTA models SAVN on the original MultiWOZ 2.0 dataset, which utilizes slot attention with the concatenated domain-slot query extracting slot specific information and value normalization on the ontologies to varying degrees, and STAR, which uses slot self-attention with the aggregated domain-slot query to model the correlations among different slots, our model obtains a JGA of 54.70% and a SA of 97.49% outperforming SAVN with a JGA of 54.52% and a SA of 97.42% , and STAR with a JGA of 54.53% and a SA of 97.38%. For the latest refined MultiWOZ 2.4 dataset, our proposed model improves the performance by a relatively larger margin comparing to the previous SOTA STAR model from a JGA of 73.62% to 75.58% and a SA of 98.87% to 98.94%. To have a better understanding, an error analysis, a discussion about the effects of different hyperparameter settings, and a case study are made and presented in

Table 1: The joint goal accuracy (JGA) and slot accuracy (SA) of different models. DDSA denotes our proposed disentangled domain-slot attention.

	Model	JGA (%)		SA (%)	
		MWZ2.0	MWZ2.4	MWZ2.0	MWZ2.4
Open vocabulary	TRADE (Wu et al., 2019)	48.93	54.97	96.92	97.58
	SOM (Kim et al., 2020)	51.72	66.78	-	98.38
	TripPy (Heck et al., 2020)	53.11	59.62	97.25	97.94
	SimpleTOD (Hosseini-Asl et al., 2020)	-	66.78	-	-
Ontology-based	SUMBT (Lee et al., 2019)	46.65	61.86	96.44	97.90
	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	97.42	98.38
	SST (Chen et al., 2020)	51.17	-	-	-
	STAR (Ye et al., 2021)	54.53	73.62	97.38	98.87
	Our model with DDSA	**54.70**	**75.58**	**97.49**	**98.94**
	Our model w/o DDSA	50.89	70.52	97.03	98.61

Appendix B. These additional results also indicate the effectiveness of our approach.

5.2 Ablation Study

A simple ablation study is performed to verify the effectiveness of our proposed disentangled domain-slot attention. As we can see in Table 1. The performance on the two datasets drops seriously when removing the proposed DDSA , which verifies the effectiveness of our proposed approach. In this case of model w/o DDSA, the domain specific and the slot specific information are extracted by feeding into the dialogue context and the domains and slots to the traditional domain and slot attention respectively, then they are concatenated and sent to the slot value matching component to perform state prediction.

6 Conclusion

In this work, we propose a model based on disentangled domain-slot attention for multi-domain dialogue state tracking to handle the correlation among different domains and slots. Unlike the conventional approach in recent mainstream models, we disentangle the query about domains and slots in a flexible and context-dependent manner. The experimental results on MultiWOZ 2.0 and MultiWOZ 2.4 datasets show that, comparing to the models based on conventional approaches of slot attention using the aggregated domain-slot pairs, our approach effectively improves the performance of multi-domain dialogue state tracking. In future works, we will investigate to utilize the proposed approach to generative models and generalize them to more complicated scenarios.

Acknowledgement

This work was supported by JSPS KAKENHI Grand Number JP22K12069 and partially supported by JSPS KAKENHI Grant Number 23K11227 and 23H03402.

Limitations

This paper shows the effectiveness of our proposed disentangled domain-slot attention mechanism in multi-domain dialogue state tracking. The limitation of this paper is that this work mainly focuses on ontology-based DST, which need a list of predefined candidate values in advance. The condition may be different in the case of generative DST since entire successive information involved in language modeling may be important for language generation. Therefore, how to tackle the problems in generated manners need to further investigate, which we intend to take up in future works.

References

Vevake Balaraman and Bernardo Magnini. 2021. Domain-aware dialogue state tracker for multi-domain dialogue systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:866–873.

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.

Guan-Lin Chao and Ian Lane. 2019. BERT-DST: Scalable End-to-End Dialogue State Tracking with Bidirectional Encoder Representations from Transformer. In *Proc. Interspeech 2019*, pages 1468–1472.

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.

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*, pages 4171–4186, Minneapolis, Minnesota. 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. 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*, volume 33, pages 20179–20191. Curran Associates, Inc.

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, Online. Association for Computational Linguistics.

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.

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.

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.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6297–6308, Red Hook, NY, USA. Curran Associates Inc.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Sarthak Mittal, Sharath Chandra Raparthy, Irina Rish, Yoshua Bengio, and Guillaume Lajoie. 2021. Compositional attention: Disentangling search and retrieval.

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, Vancouver, Canada. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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 (EMNLP)*, pages 2780–2786, Brussels, Belgium. Association for Computational Linguistics.

Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. 2021.

Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135.

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.

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.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432, Metz, France. Association for Computational Linguistics.

Jason D. Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Comput. Speech Lang.*, 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.

Longfei Yang, Jiyi Li, Sheng Li, and Takahiro Shinozaki. 2022. Multi-domain dialogue state tracking with top-k slot self attention. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 231–236, Edinburgh, UK. Association for Computational Linguistics.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2022. 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*, pages 1598–1608.

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.

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, Melbourne, Australia. Association for Computational Linguistics.

A Experimental Settings

The dialogue context encoder BERT$_{context}$ in this work is a pre-trained BERT-base-uncased model, which has 12 layers with 768 hidden units and 12 self-attention heads. We also employ another BERT-base-uncased model as the domain, slot and value encoder BERT$_{dsv}$. For the proposed disentangled domain-slot attention, the number of heads of domains N_d and that of slots N_s in disentangled domain-slot attention are hyperparameters and investigated in the experiments. The dimension k_{ddsa} in it is set to 768. Adam optimizer is adopted with a batch size of 8, which trains the model with a learning rate of 4e-5 for the encoder and 1e-4 for other parts. The hyperparameters are selected from the best-performing model over the validation set. We use a dropout with a probability of 0.1 on the dialogue history during training. The ground-truth states at previous turns are involved in the input during training. The previously predicted states are used as part of the input when inferring.

B Supplementary Results

B.1 Effects of Different Hyperparameter Settings

To investigate the effects of different hyperparameter settings, Table 2 presents the results of using different numbers of heads N_d for domain query and that N_s for slot query in the DDSA component in our model. It can be found that the model achieves the best performance when the number of heads for domain $N_d = 16$ and that for slot $N_s = 32$ in the experiment. These hyperparameters are selected by tuning on the validation set.

B.2 Error Analysis

An error analysis of each slot for the previous SOTA model STAR and our model on MultiWOZ 2.4 is shown in Figure 2, in which the lower the better. It can be observed that the error rates of several $name$ and $area$-related slots are improved significantly. Specifically, the performance of $restaurant-name$, $hotel-type$, $hotel-area$, $attraction - area$ and $hotel - bookstay$ are improved to a relatively large margin.

B.3 Case Study

A case study below demonstrates some cases in MultiWOZ 2.4 dataset. Table 3 presents three dialogue episodes and the predicted dialogue states by the previous SOTA STAR and our proposed model. It can be found that, for the first example, the system recommends "downing college" to the user's request for an attraction. Although STAR captures the adjective phrase, the value for the slot $attraction - name$ is not the referencing object "all saints church". Since there is a full slot self-attention is applied to the concatenated domain-slot query specific information, the mistake may be introduced from other domain-slot specific represen-

Table 2: The results of our models with different numbers of heads N_d for domain query and that of N_s for slot query on MultiWOZ 2.4 dataset.

N_d	N_s	JGA
4	8	71.58
4	16	71.14
8	16	72.8
8	32	74.28
16	16	74.47
16	32	75.58
16	64	74.08

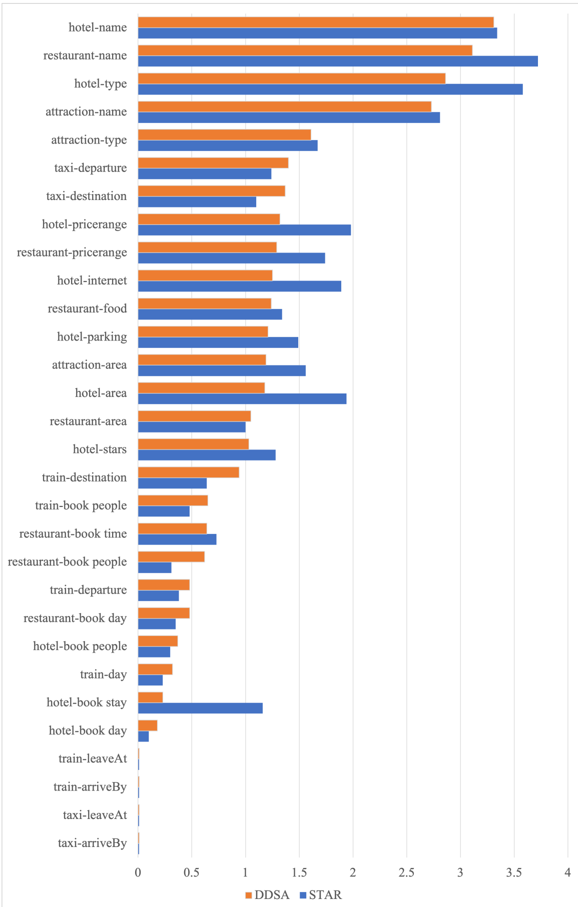


Figure 2: The error rate per slot of STAR and our model based on proposed DDSA on MultiWOZ 2.4 dataset.

Table 3: The dialogue state prediction for three dialogue episodes in the MultiWOZ 2.4 dataset. We omit some slots and values for simplification.

Dialogue context	STAR	DDSA
SYS: I recommend downing college. **USR**: How far is it from the all saints church?	attraction-name=all saints church	attraction-name=downing college
SYS: I completed your booking. Your reference number is 35w3xedl. Is there anything else I could do to help? **USR**: Yes, I also need to verify that this hotel is in the east area of the town.	hotel-area=none	hotel-area=east
SYS: I have over 20 different options for you, was there a certain area or price range you would like me to find for you? **USR**: Let's see what is available cheap, same area as the restaurant makes most sense but I am open to any area.	hotel-area=south	hotel-area=do not care

tations. In the second case, the user would like to confirm the asked hotel in the east area while STAR fails to get the point. In the third case, the user is open to any area. But STAR still overestimated the correlation between $hotel$ and the previously mentioned $restaurant$. Our model successfully predicts the dialogue states for it.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitation

☐ 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?
Abstract and Introduction

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

B ☑ Did you use or create scientific artifacts?

Section 3; cited in References

☑ B1. Did you cite the creators of artifacts you used?
Section 3; cited in References

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 3; cited in References

☑ 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)?
Section 3; cited in References

☑ 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?
Section 3; cited in References

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3; cited in References

☑ 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 3 states that we take the step as same in other works for consistency.

C ☒ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
No response.

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.