Controllable User Dialogue Act Augmentation for Dialogue State Tracking

Chun-Mao Lai*   Ming-Hao Hsu*   Chao-Wei Huang   Yun-Nung Chen
National Taiwan University, Taipei, Taiwan
{b09901186, b09502138}@ntu.edu.tw
f07922069@csie.ntu.edu.tw   y.v.chen@ieee.org

Abstract

Prior work has demonstrated that data augmentation is useful for improving dialogue state tracking. However, there are many types of user utterances, while the prior method only considered the simplest one for augmentation, raising the concern about poor generalization capability. In order to better cover diverse dialogue acts and control the generation quality, this paper proposes controllable user dialogue act augmentation (CUDA-DST) to augment user utterances with diverse behaviors. With the augmented data, different state trackers gain improvement and show better robustness, achieving the state-of-the-art performance on MultiWOZ 2.1.[1]

1 Introduction

Dialogue state tracking (DST) serves as a backbone of task-oriented dialogue systems (Chen et al., 2017), where it aims at keeping track of user intents and associated information in a conversation. The dialogue states encapsulate the required information for the subsequent dialogue components. Hence, an accurate DST module is crucial for a dialogue system to perform successful conversations.

Recently, we have seen tremendous improvement on DST, mainly due to the curation of large datasets (Budzianowski et al., 2018; Eric et al., 2020; Rastogi et al., 2020) and many advanced models. They can be broadly categorized into 3 types: span prediction, question answering, and generation-based models. The question answering models define natural language questions for each slot to query the model for the corresponding values (Gao et al., 2020; Li et al., 2021). Wu et al. (2019) proposed TRADE to perform zero-shot transfer between multiple domains via slot-value embeddings and a state generator. SimpleTOD (Hosseini-Asl et al., 2020) combines all components in a task-oriented dialogue system with a pre-trained language model. Recently, TripPy (Heck et al., 2020) categorizes value prediction into 7 types, and designs different prediction strategies for them. This paper focuses on generalized augmentation covering all categories.

Another research line leverages data augmentation techniques to improve performance (Song et al., 2021; Yin et al., 2020; Summerville et al., 2020; Kim et al., 2021). Most prior work used simple augmentation techniques such as word insertion and state value substitution. With recent advances in pre-trained language models (Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2020), generation-based augmentation has been proposed (Kim et al., 2021; Li et al., 2020). These methods have demonstrated impressive improvement and zero-shot adaptability (Yoo et al., 2020; Campagna et al., 2020), while our work focuses on data augmentation with in-domain data.

The closest work is CoCo (Li et al., 2020), a framework that generates user utterances given augmented dialogue states. The examples are shown in Figure 1, where the main differences between CoCo and ours are that 1) CoCo only augments user utterances in slot and value levels, but dialogue acts and domains are fixed, making augmented data limited. Our method can augment reasonable user utterances with diverse dialogue acts and domain switching scenarios. 2) Boolean slots and referred slots are not handled by CoCo due to its higher complexity, while our approach can handle all types of values for better generalization.

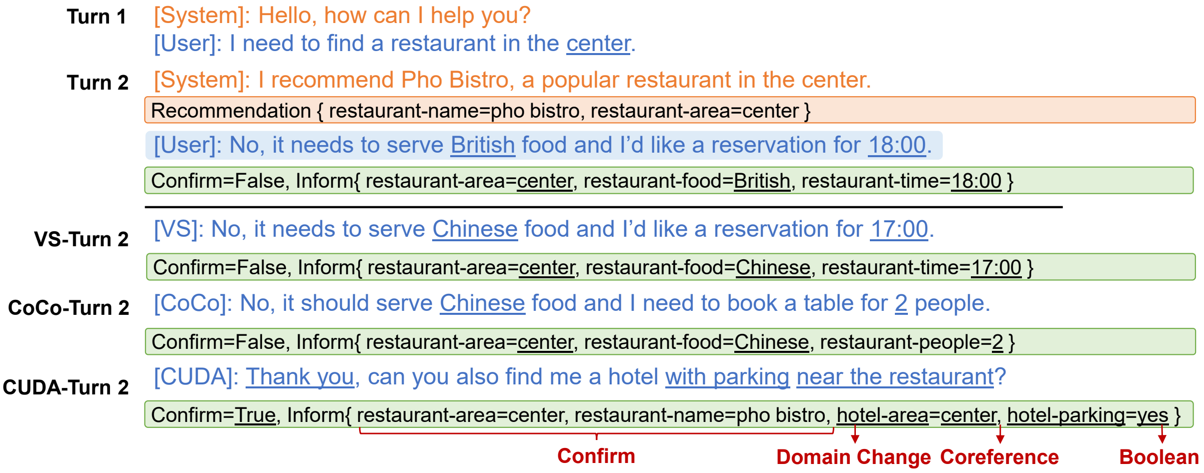
This paper proposes **CUDA-DST** (**C**ontrollable **U**ser **D**ialogue **A**ct augmentation), a generalized framework of generation-based augmentation for improving DST. Our contribution is 2-fold:

- We present CUDA which generates diverse user utterances via controllable user dialogue acts augmentation.
- Our augmented data helps most DST mod-

[*] Equal contribution.

[1] The source code is available at https://github.com/MiuLab/CUDA-DST.

Proceedings of the SIGdial 2022 Conference, pages 53–61
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022 ©2022 Association for Computational Linguistics



Turn 1
[System]: Hello, how can I help you?
[User]: I need to find a restaurant in the center.
Turn 2
[System]: I recommend Pho Bistro, a popular restaurant in the center.
Recommendation { restaurant-name=pho bistro, restaurant-area=center }
[User]: No, it needs to serve British food and I'd like a reservation for 18:00.
Confirm=False, Inform{ restaurant-area=center, restaurant-food=British, restaurant-time=18:00 }
VS-Turn 2
[VS]: No, it needs to serve Chinese food and I'd like a reservation for 17:00.
Confirm=False, Inform{ restaurant-area=center, restaurant-food=Chinese, restaurant-time=17:00 }
CoCo-Turn 2
[CoCo]: No, it should serve Chinese food and I need to book a table for 2 people.
Confirm=False, Inform{ restaurant-area=center, restaurant-food=Chinese, restaurant-people=2 }
CUDA-Turn 2
[CUDA]: Thank you, can you also find me a hotel with parking near the restaurant?
Confirm=True, Inform{ restaurant-area=center, restaurant-name=pho bistro, hotel-area=center, hotel-parking=yes }
Confirm
Domain Change
Coreference
Boolean

Figure 1: Augmented user utterances with the associated user dialogue acts and states from three methods.

els improve their performance. Specifically, CUDA-augmented TripPy model achieves the state-of-the-art result on MultiWOZ 2.1.

2 Controllable User Dialogue Act Augmentation (CUDA)

The goal of our method is to augment more and diverse user utterances that fit the dialogue context, and then the augmented data can help DST models learn better. More formally, given a system utterance U_t^{sys} in the turn t and dialogue history H_{t-1} before this turn, our approach focuses on augmenting a user dialogue act and state, $\hat{A}_t$, and generating the corresponding user utterance $\hat{U}_t^{\text{usr}}$. Note that each user utterance can be augmented.

To achieve this goal, we propose CUDA with three components illustrated in Figure 2: 1) a user dialogue act generation process for producing $\hat{A}_t$, 2) a user utterance generator for producing $\hat{U}_t^{\text{usr}}$, and 3) a state match filtering process.

2.1 User Dialogue Act Generation

Considering that a user dialogue act represents the core meaning of the user's behavior (Goo and Chen, 2018; Yu and Yu, 2021), we focus on simulating reasonable user dialogue acts given the system context for data augmentation. After analyzing task-oriented user utterances, user behaviors contain the following user dialogue acts:

1. **Confirm**: The system provides recommendation to the user, and the user confirms if accepting the recommended item.
2. **Reply**: The system asks for a user-desired value of the slots, and the user replies the corresponding value.
3. **Inform**: The user directly informs the desired slot values to the system.

Heck et al. (2020) designed their dialogue state tracker that tackle utterances with different dialogue acts in different ways and achieved good performance, implying that different dialogue acts contain diverse behaviors in the interactions. To augment more diverse user utterances, we introduce a random process for each user dialogue act. Unlike the prior work CoCo that did not generate utterance whose dialogue act different from the original one, our design is capable of simulating diverse behaviors for better augmentation illustrated in Figure 2.

Confirm When the system provides recommendations, our augmented user behavior has a probability of P_{confirm} to accept the recommended values. When the user confirms the recommendation, the suggested slot values are added to the augmented user dialogue state $\hat{A}_t$ as shown in Figure 1. In the example, the augmented user dialogue act is to confirm the suggested restaurant, and then includes it in the state (restaurant-name=pho bistro, restaurant-area=center).

Reply When the system requests a constraint for a specific slot, e.g. "*which area do you prefer?*", the user has a probability of P_{reply} to give the value of the requested slot. P_{reply} may not be 1, because users sometimes revise their previous requests without providing the asked information.

Inform In anytime of the conversation, the user can provide the desired slot values to convey his/her preference. As shown in the original user utterance of Figure 1, the user rejects the recommendation

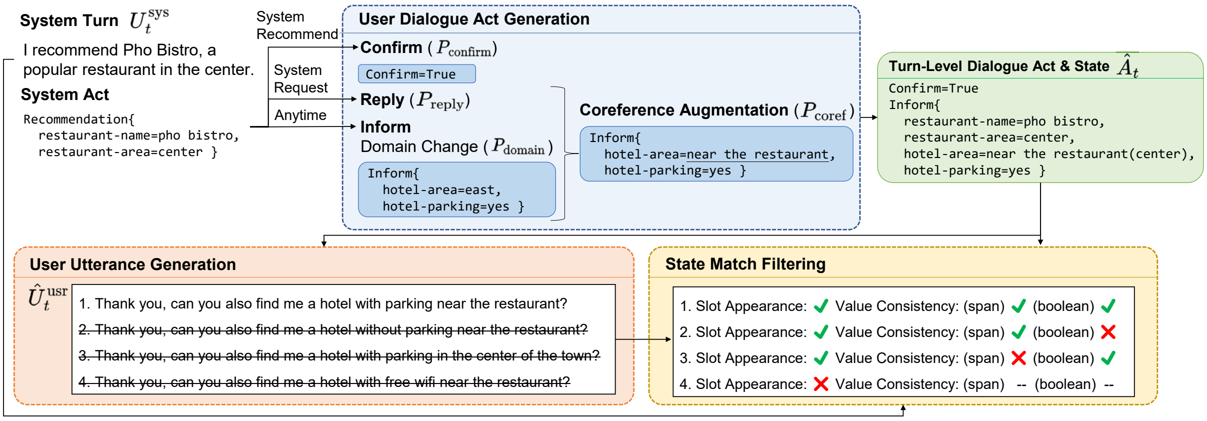

System Turn U_t^{sys}
I recommend Pho Bistro, a popular restaurant in the center.
System Act
Recommendation{ restaurant-name=pho bistro, restaurant-area=center }
System Recommend
System Request
Anytime
User Dialogue Act Generation
Confirm (P_{confirm})
Confirm=True
Reply (P_{reply})
Inform
Domain Change (P_{domain})
Inform{ hotel-area=east, hotel-parking=yes }
Coreference Augmentation (P_{coref})
Inform{ hotel-area=near the restaurant, hotel-parking=yes }
Turn-Level Dialogue Act & State $\hat{A}_t$
Confirm=True
Inform{ restaurant-name=pho bistro, restaurant-area=center, hotel-area=near the restaurant(center), hotel-parking=yes }
User Utterance Generation
$\hat{U}_t^{\text{usr}}$
1. Thank you, can you also find me a hotel with parking near the restaurant?
~~2. Thank you, can you also find me a hotel without parking near the restaurant?~~
~~3. Thank you, can you also find me a hotel with parking in the center of the town?~~
~~4. Thank you, can you also find me a hotel with free wifi near the restaurant?~~
State Match Filtering
1. Slot Appearance: ✓ Value Consistency: (span) ✓ (boolean) ✓
2. Slot Appearance: ✓ Value Consistency: (span) ✓ (boolean) ✗
3. Slot Appearance: ✓ Value Consistency: (span) ✗ (boolean) ✓
4. Slot Appearance: ✗ Value Consistency: (span) -- (boolean) --

Figure 2: The overview of the proposed CUDA augmentation process.

and then directly informs the additional constraints (food and time). The number of additional informed values is randomly chosen, and then the slots and values are randomly sampled from the pre-defined ontology and dictionary. Note that the confirmed and replied information cannot be changed during additional informing. Considering that a user may change the domain within the dialogue, our algorithm allows the user to change the domain with a probability of P_{domain}, and then the informed slots and values need to be sampled from the new domain's dictionary. The new domain is selected randomly from all the other domains.

Coreference Augmentation In the generated user dialogue act and state, all informed slot values are from the pre-defined dictionary. However, it is natural for a user to refer the previously mentioned information, e.g., "*I am looking for a taxi that can arrive by the time of my reservation*". To further enhance the capability of handling coreference, our algorithm has a probability of P_{coref} to switch the slot value from the generated user dialogue state. Since not all slots can be referred, we define a coreference list containing all referable slots and the corresponding referring phrases, e.g., "*the same area as*" listed in Appendix A.

With the generated user dialogue acts and the system action, we form the corresponding turn-level dialogue act and state based on the confirmed suggestions and referred slot values as shown in the green block of Figure 2.

2.2 User Utterence Generation

To generate the user utterance associated with the augmented user dialogue act and state, we adopt a pre-trained T5 (Raffel et al., 2020) and fine-tune it on the MultiWOZ dataset by a language modeling objective formulated below:

$$\mathcal{L}_{\text{gen}} = -\sum_{k=1}^{n_t} \log p_\theta(U_{t,k}^{\text{usr}} \mid U_{t,<k}^{\text{usr}}, U_t^{\text{sys}}, H_{t-1}, A_t),$$

where $U_{t,k}^{\text{usr}}$ denotes the k-th token in the user utterance, H_{t-1} represents the all dialogue history before turn t, and A_t is the user dialogue act and state in the t-th turn. With the trained generator, we can generate the augmented user utterance by inputting the augmented user dialogue act and state $\hat{A}_t$ as shown in the green block of Figure 2. In decoding, we apply beam search so that we can augment diverse utterances for improving DST.

2.3 State Match Filtering

To make sure the generated user utterance well reflects its dialogue state, we propose two modules to check the state matching: a *slot appearance* classifier and a *value consistency* filter, where the former checks if the given slots are included and the latter focuses on ensuring the value consistency between dialogue states and user utterances.

Slot Appearance Following Li et al., we employ a BERT-based multi-label classification model to predict whether a slot appears in the given t-th turn. The augmented user utterances are eliminated if they do not contain all slots in the user dialogue state predicted by the model.

Value Consistency The slot values can be categorized into: 1) span-based, 2) boolean, and 3) *dontcare* values. It is naive to check if the span-based values are mentioned in the utterances, but boolean and *dontcare* values cannot be easily identified. To handle the slots with boolean and *dontcare* values, we propose two slot-gate classifiers

Dataset	CUDA	MultiWOZ
Span	100.00	64.61
Confirm (True)	5.27	5.84
Confirm (False)	0.44	0.32
Dontcare	0.67	2.46
Coreference	8.15	3.70
Multi-domain	13.10	24.48
#Turns	*54,855*	*69,673*

Table 1: Slot distribution in user utterances (%).

MultiWOZ	TripPy	TRADE	SimpleTOD
Original	57.72	44.08	49.19
VS	59.48	43.76	**50.50**
CoCo	60.46	43.53	50.25
CUDA	$61.28^{\dagger}$	$\mathbf{44.86}^{\dagger}$	50.14
CUDA (-*coref*)	$\mathbf{62.93}^{\dagger}$	42.98	49.64

Table 2: Joint goal accuracy on MultiWOZ 2.1 (%). † indicates the significant improvement over all baselines with $p < 0.05$.

motivated by Heck et al. (2020). Each boolean slot, e.g. *internet* or *parking*, is assigned to one of the classes in $C_{\text{bool}} = \{none, dontcare, yes, no\}$, while other slots are assigned to one of the classes in $C_{\text{span}} = \{none, dontcare, value\}$, where *value* indicates the span-based value. Then for all slots classified as span-based value, we check if all associated values are mentioned in the generated utterance. In addition, we use the coreference keywords, e.g., *same area*, to handle the coreference cases. We apply BERT (Devlin et al., 2019) to encode the t-th turn in a dialogue as:

$$R_t^{\text{CLS}} = \text{BERT}([\text{CLS}] \oplus U_t^{\text{sys}} \oplus [\text{SEP}] \oplus U_t^{\text{usr}} \oplus [\text{SEP}]),$$

where R_t^{CLS} denotes the output of the [CLS] token, which can be considered as the summation of the turn t. We then obtain the probability of the value types as

$$p_{s,t}^{\text{bool}} = \text{softmax}(W_s^{\text{bool}} \cdot R_t^{\text{CLS}} + b_s^{\text{bool}}) \in \mathbb{R}^4,$$

for each boolean slots, and

$$p_{s,t}^{\text{span}} = \text{softmax}(W_s^{\text{span}} \cdot R_t^{\text{CLS}} + b_s^{\text{span}}) \in \mathbb{R}^3,$$

for each span-based slots. Our multi-task BERT-based slot-gate classifier is trained with the cross entropy loss.

The neural-based filters are trained on the original MultiWOZ data, and the prediction performance in terms of slots (for both appearance and value consistency) is 92.9% in F1 evaluated on the development set. In our CUDA framework, we apply the trained filters to ensure the quality of the augmented user utterances as shown in Figure 2.

3 Experiments

To evaluate if our augmented data is beneficial for improving DST models, we perform three popular trackers, TRADE (Wu et al., 2019), SimpleTOD (Hosseini-Asl et al., 2020), and TripPy (Heck et al., 2020), on MultiWOZ 2.1 (Eric et al., 2020).

3.1 Experimental Setting

Our CUDA generator is trained on the training set of MultiWOZ 2.3 (Han et al., 2020) due to its additional *coreference* labels. Note that all dialogues are the same as MultiWOZ 2.1. We then generate the augmented dataset for the training set of MultiWOZ 2.1 for fair comparison with the prior work. The predifined slot-value dictionary is taken from CoCo's *out-of-domain* dictionary and the defined coreference list is shown in Appendix A.

In user dialogue act generation, the parameters are set as $(P_{\text{confirm}}, P_{\text{reply}}, P_{\text{domain}}, P_{\text{coref}}) = (0.7, 0.9, 0.8, 0.6)$, which can be flexibly adjusted to simulate different user behaviors. We report the distribution of slot types in our augmented data and the original MultiWOZ data in Table 1, where it can be found that our augmented slots cover diverse slot types and the distribution is reasonably similar to the original MultiWOZ. Different from the prior work, CoCo, which only tackled the span-based slots, our augmented data may better reflect the natural conversational interactions. Additionally, we perform CUDA with $P_{\text{coref}} = 0$ to check the impact of coreference augmentation.

We train three DST models on the augmented data and evaluate the results using joint goal accuracy. The compared augmentation baselines include value substitution (VS) and CoCo (Li et al., 2020) with the same setting.

3.2 Effectiveness of CUDA-Augmented Data

Table 2 shows that CUDA significantly improves TripPy and TRADE results by 3.6% and 0.8% respectively on MultiWOZ, and even outperforms the prior work CoCo. In addition, our CUDA augmentation process has 78% success rate, while CoCo only has 57%, demonstrating the efficiency of our augmentation method and the great data utility. Interestingly, CUDA without *coreference* achieves slightly better performance for TripPy while the performance of TRADE and SimpleTOD degrade,

CoCo+(rare)	TripPy	TRADE	SimpleTOD
Original	28.38	16.65	19.20
VS	39.42	16.42	26.26
CUDA	**48.83**	**17.79**	**29.32**
CUDA (*-coref*)	48.67	16.80	28.66
CoCo	*56.50*	*18.01*	*30.60*

Table 3: Joint goal accuracy on CoCo+ (rare) (%).

achieving the new state-of-the-art performance on MultiWOZ 2.1. The probable reason is that TripPy already handles coreference very well via its refer classification module, so augmenting coreference cases may not help it a lot. In contrast, other generative models (TRADE and SimpleTOD) can benefit more from our augmented coreference cases. Another reason may be the small distribution of coreference slots in MultiWOZ shown as Table 1, implying that augmented data with too many coreference slots does not align well with the original distribution and hurts the performance.

3.3 Robustness to Rare Cases

We also evaluate our models on *CoCo+ (rare)*[2], a test set generated by CoCo's algorithm (Li et al., 2020), to examine model robustness under rare scenarios. Table 3 presents the results on CoCo+ (rare), which focuses rare cases for validating the model's robustness. It is clear that the model trained on our augmented data shows better generalization compared with the one trained on the original MultiWOZ data, demonstrating the effectiveness on improving robustness of DST models. The performance of CoCo is listed as reference, because comparing with its self-generated data is unfair.

3.4 Slot Performance Analysis

To further investigate the efficacy for each slot type, Figure 3 presents its performance gain on TripPy. Comparing with CoCo, CUDA improves more on *informed*, *refer*, and *dontcare* slots. It implies that CUDA augments diverse user dialogue acts for helping *informed* and *refer*, and the proposed slot-gate can better ensure value consistency for improving *dontcare* slots, even though they are rare cases in MultiWOZ. Our model can also keep the same performance for frequent *span* slots, demonstrating great generalization capability across diverse slot types from our controllable augmentation. The qualitative study can be found in Appendix B.

[2] CoCo+ (rare) applies *CoCo* and *value substitution (VS)* with a *rare* slot-combination dictionary.

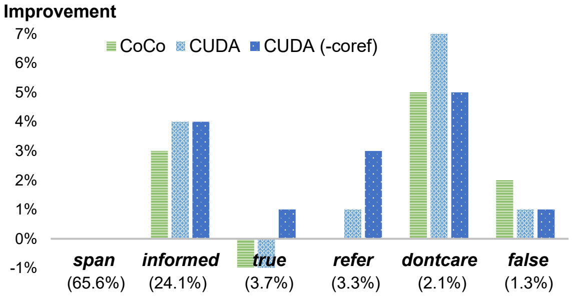


Figure 3: Performance gain across slots on TripPy.

4 Conclusion

We introduce a generalized data augmentation method for DST by utterance generation with controllable user dialogue act augmentation. Experiments show that our approach improves results of multiple state trackers and achieves state-of-the-art performance on MultiWOZ 2.1. Further study demonstrates that trackers' robustness and generalization capabilities can be improved by diverse generation covering different user behaviors.

Acknowledgements

We thank reviewers for their insightful comments. This work was financially supported from the Young Scholar Fellowship Program by Ministry of Science and Technology (MOST) in Taiwan, under Grants 111-2628-E-002-016 and 111-2634-F-002-014.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 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.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Yun-Nung Chen, Asli Celikyilmaz, and Dilek Hakkani-Tur. 2017. Deep learning for dialogue systems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 8–14.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. ACL.

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 *the 12th Language Resources and Evaluation Conference*, pages 422–428. European Language Resources Association.

Shuyang Gao, Sanchit Agarwal, Di Jin, Tagyoung Chung, and Dilek Hakkani-Tur. 2020. From machine reading comprehension to dialogue state tracking: Bridging the gap. In *the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 79–89. ACL.

Chih-Wen Goo and Yun-Nung Chen. 2018. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 735–742. IEEE.

Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Wei Peng, and Minlie Huang. 2020. MultiWOZ 2.3: A multi-domain taskoriented dataset enhanced with annotation corrections and co-reference annotation. *arXiv preprint arXiv:2010.05594*.

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 *the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44. ACL.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191.

Sungdong Kim, Minsuk Chang, and Sang-Woo Lee. 2021. NeuralWOZ: Learning to collect task-oriented dialogue via model-based simulation. In *the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 3704–3717. ACL.

Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2020. CoCo: Controllable counterfactuals for evaluating dialogue state trackers. In *International Conference on Learning Representations*.

Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian McAuley. 2021. Zero-shot generalization in dialog state tracking through generative question answering. In *the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1063–1074. ACL.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Xiaohui Song, Liangjun Zang, and Songlin Hu. 2021. Data augmentation for copy-mechanism in dialogue state tracking. In *International Conference on Computational Science*, pages 736–749. Springer.

Adam Summerville, Jordan Hashemi, James Ryan, and William Ferguson. 2020. How to tame your data: Data augmentation for dialog state tracking. In *the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 32–37. ACL.

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 *the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819. ACL.

Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dialog state tracking with reinforced data augmentation. In *the AAAI Conference on Artificial Intelligence*, volume 34, pages 9474–9481.

Kang Min Yoo, Hanbit Lee, Franck Dernoncourt, Trung Bui, Walter Chang, and Sang-goo Lee. 2020. Variational hierarchical dialog autoencoder for dialog state tracking data augmentation. In *the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3406–3425. ACL.

Dian Yu and Zhou Yu. 2021. Midas: A dialog act annotation scheme for open domain humanmachine spoken conversations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1103–1120.

A Reproducibility

Our CUDA generator is trained on the training set of MultiWOZ 2.3 (Han et al., 2020) due to its additional *coreference* labels. Note that all dialogues are the same as MultiWOZ 2.1. We then generate the augmented dataset using CUDA for the training set of MultiWOZ 2.1 for fair comparison with the prior work. The predifined slot-value dictionary is taken from CoCo's *out-of-domain* dictionary shown in Table 4 and the defined coreference list is shown in Table 5.

B Qualitative Study

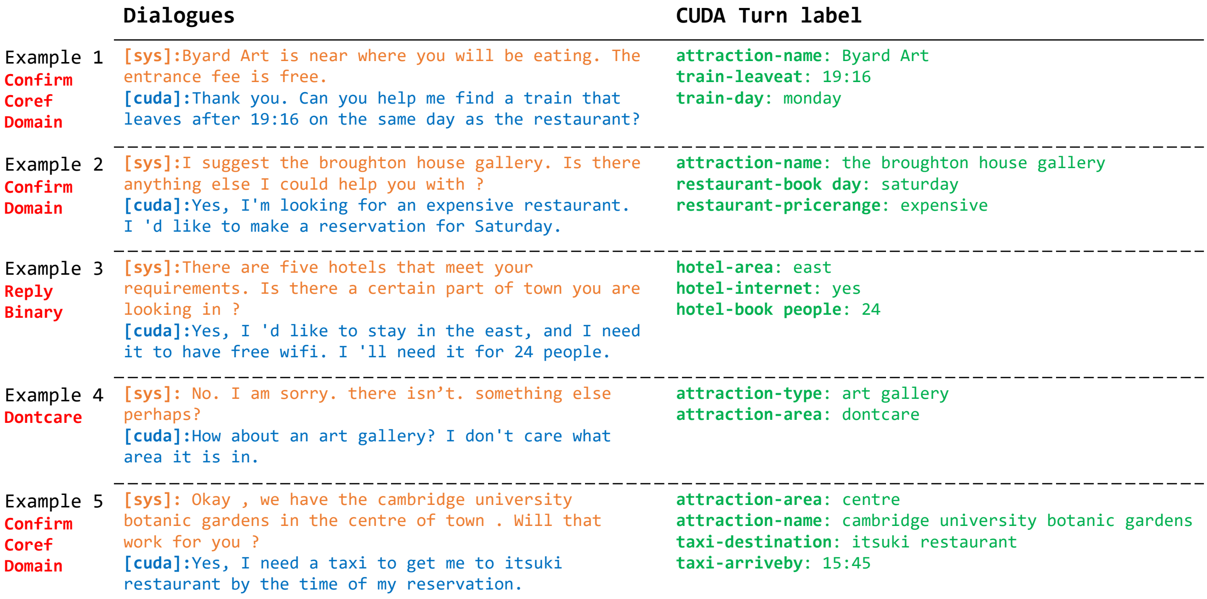
The augmented data samples are shown in Figure 4. It can be found that the augmented user utterances can fluently switch the domain and include associated slot values that are aligned well with the dialogue states.

Slot Name	**Possible Values**
hotel-internet†	[‘yes’, ‘no’, ‘dontcare’]
hotel-type	[‘hotel’, ‘guesthouse’]
hotel-parking†	[‘yes’, ‘no’, ‘dontcare’]
hotel-price	[‘moderate’, ‘cheap’, ‘expensive’]
hotel-day	[‘march 11th’, ‘march 12th’, ‘march 13th’, ‘march 14th’, ‘march 15th’, ‘march 16th’, ‘march 17th’, ‘march 18th’, ‘march 19th’, ‘march 20th’]
hotel-people	[‘20’, ‘21’, ‘22’, ‘23’, ‘24’, ‘25’, ‘26’, ‘27’, ‘28’, ‘29’]
hotel-stay	[‘20’, ‘21’, ‘22’, ‘23’, ‘24’, ‘25’, ‘26’, ‘27’, ‘28’, ‘29’]
hotel-area	[‘south’, ‘north’, ‘west’, ‘east’, ‘centre’, ‘dontcare’]
hotel-stars	[‘0’, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘dontcare’]
hotel-name	[‘moody moon’, ‘four seasons hotel’, ‘knights inn’, ‘travelodge’, ‘jack summer inn’, ‘paradise point resort’]
restaurant-area	[‘south’, ‘north’, ‘west’, ‘east’, ‘centre’, ‘dontcare’]
restaurant-food	[‘asian fusion’, ‘burger’, ‘pasta’, ‘ramen’, ‘taiwanese’, ‘dontcare’]
restaurant-price	[‘moderate’, ‘cheap’, ‘expensive’, ‘dontcare’]
restaurant-name	[‘buddha bowls’, ‘pizza my heart’, ‘pho bistro’, ‘sushiya express’, ‘rockfire grill’, ‘itsuki restaurant’]
restaurant-day	[‘monday’, ‘tuesday’, ‘wednesday’, ‘thursday’, ‘friday’, ‘saturday’, ‘sunday’]
restaurant-people	[‘20’, ‘21’, ‘22’, ‘23’, ‘24’, ‘25’, ‘26’, ‘27’, ‘28’, ‘29’]
restaurant-time	[‘19:01’, ‘18:06’, ‘17:11’, ‘19:16’, ‘18:21’, ‘17:26’, ‘19:31’, ‘18:36’, ‘17:41’, ‘19:46’, ‘18:51’, ‘17:56’, ‘7:00 pm’, ‘6:07 pm’, ‘5:12 pm’, ‘7:17 pm’, ‘6:17 pm’, ‘5:27 pm’, ‘7:32 pm’, ‘6:37 pm’, ‘5:42 pm’, ‘7:47 pm’, ‘6:52 pm’, ‘5:57 pm’, ‘11:00 am’, ‘11:05 am’, ‘11:10 am’, ‘11:15 am’, ‘11:20 am’, ‘11:25 am’, ‘11:30 am’, ‘11:35 am’, ‘11:40 am’, ‘11:45 am’, ‘11:50 am’, ‘11:55 am’]
restaurant-food	[‘asian fusion’, ‘burger’, ‘pasta’, ‘ramen’, ‘taiwanese’, ‘dontcare’]
taxi-arrive	[‘17:26’, ‘19:31’, ‘18:36’, ‘17:41’, ‘19:46’, ‘18:51’, ‘17:56’, ‘7:00 pm’, ‘6:07 pm’, ‘5:12 pm’, ‘7:17 pm’, ‘6:17 pm’, ‘5:27 pm’, ‘11:30 am’, ‘11:35 am’, ‘11:40 am’, ‘11:45 am’, ‘11:50 am’, ‘11:55 am’]
taxi-leave	[‘19:01’, ‘18:06’, ‘17:11’, ‘19:16’, ‘18:21’, ‘7:32 pm’, ‘6:37 pm’, ‘5:42 pm’, ‘7:47 pm’, ‘6:52 pm’, ‘5:57 pm’, ‘11:00 am’, ‘11:05 am’, ‘11:10 am’, ‘11:15 am’, ‘11:20 am’, ‘11:25 am’]
taxi-depart	[‘moody moon’, ‘four seasons hotel’, ‘knights inn’, ‘travelodge’, ‘jack summer inn’, ‘paradise point resort’]
taxi-dest	[‘buddha bowls’, ‘pizza my heart’, ‘pho bistro’, ‘sushiya express’, ‘rockfire grill’, ‘itsuki restaurant’]
train-arrive	[‘17:26’, ‘19:31’, ‘18:36’, ‘17:41’, ‘19:46’, ‘18:51’, ‘17:56’, ‘7:00 pm’, ‘6:07 pm’, ‘5:12 pm’, ‘7:17 pm’, ‘6:17 pm’, ‘5:27 pm’, ‘11:30 am’, ‘11:35 am’, ‘11:40 am’, ‘11:45 am’, ‘11:50 am’, ‘11:55 am’]
train-leave	[‘19:01’, ‘18:06’, ‘17:11’, ‘19:16’, ‘18:21’, ‘7:32 pm’, ‘6:37 pm’, ‘5:42 pm’, ‘7:47 pm’, ‘6:52 pm’, ‘5:57 pm’, ‘11:00 am’, ‘11:05 am’, ‘11:10 am’, ‘11:15 am’, ‘11:20 am’, ‘11:25 am’]
train-depart	[‘gilroy’, ‘san martin’, ‘morgan hill’, ‘blossom hill’, ‘college park’, ‘santa clara’, ‘lawrence’, ‘sunnyvale’]
train-dest	[‘mountain view’, ‘san antonio’, ‘palo alto’, ‘menlo park’, ‘hayward park’, ‘san mateo’, ‘broadway’, ‘san bruno’]
train-day	[‘march 11th’, ‘march 12th’, ‘march 13th’, ‘march 14th’, ‘march 15th’, ‘march 16th’, ‘march 17th’, ‘march 18th’, ‘march 19th’, ‘march 20th’]
train-people	[‘20’, ‘21’, ‘22’, ‘23’, ‘24’, ‘25’, ‘26’, ‘27’, ‘28’, ‘29’]
attraction-area	[‘south’, ‘north’, ‘west’, ‘east’, ‘centre’, ‘dontcare’]
attraction-name	[‘grand canyon’, ‘golden gate bridge’, ‘niagara falls’, ‘kennedy space center’, ‘pike place market’, ‘las vegas strip’]
attraction-type	[‘historical landmark’, ‘aquaria’, ‘beach’, ‘castle’, ‘art gallery’, ‘dontcare’]

Table 4: The pre-defined slot-value dictionary, where † indicates a binary slot.

Slot Name	Referred Slot Name	Referred Key Value
hotel-price	*restaurant-price*	[‘same’, ‘same price’, ‘same price range’]
hotel-day	*train-day*	[‘same’, ‘same day’]
	restaurant-day	[‘same’, ‘same day’]
hotel-people	*train-people*	[‘same’, ‘same group’, ‘same party’]
	restaurant-people	[‘same’, ‘same group’, ‘same party’]
hotel-area	*restaurant-area*	[‘same’, ‘same area’, ‘same part’, ‘near the restaurant’]
	attraction-area	[‘same’, ‘same area’, ‘same part’, ‘near the attraction’]
restaurant-area	*hotel-area*	[‘same’, ‘same area’, ‘same part’, ‘near the hotel’]
	attraction-area	[‘same’, ‘same area’, ‘same part’, ‘near the attraction’]
restaurant-price	*hotel-price*	[‘same’, ‘same price’, ‘same price range’]
restaurant-day	*train-day*	[‘same’, ‘same day’]
	hotel-day	[‘same’, ‘same day’]
restaurant-people	*train-people*	[‘same’, ‘same group’, ‘same party’]
	hotel-people	[‘same’, ‘same group’, ‘same party’]
taxi-depart	*hotel-name*	[‘the hotel’]
	restaurant-name	[‘the restaurant’]
	attraction-name	[‘the attraction’]
taxi-dest	*hotel-name*	[‘the hotel’]
	restaurant-name	[‘the restaurant’]
	attraction-name	[‘the attraction’]
taxi-arrive	*restaurant-time*	[‘the time of my reservation’, ‘the time of my booking’]
train-day	*restaurant-day*	[‘same’, ‘same day’]
	hotel-day	[‘same’, ‘same day’]
train-people	*restaurant-people*	[‘same’, ‘same group’, ‘same party’]
	hotel-people	[‘same’, ‘same group’, ‘same party’]
attraction-area	*hotel-area*	[‘same’, ‘same area’, ‘same part’, ‘near the hotel’]
	restaurant-area	[‘same’, ‘same area’, ‘same part’, ‘near the restaurant’]

Table 5: The coreference list. The slots that is not referable will not be displayed in the above table.



	Dialogues	CUDA Turn label
Example 1 Confirm Coref Domain	[sys]:Byard Art is near where you will be eating. The entrance fee is free. [cuda]:Thank you. Can you help me find a train that leaves after 19:16 on the same day as the restaurant?	attraction-name: Byard Art train-leaveat: 19:16 train-day: monday
Example 2 Confirm Domain	[sys]:I suggest the broughton house gallery. Is there anything else I could help you with ? [cuda]:Yes, I'm looking for an expensive restaurant. I 'd like to make a reservation for Saturday.	attraction-name: the broughton house gallery restaurant-book day: saturday restaurant-pricerange: expensive
Example 3 Reply Binary	[sys]:There are five hotels that meet your requirements. Is there a certain part of town you are looking in ? [cuda]:Yes, I 'd like to stay in the east, and I need it to have free wifi. I 'll need it for 24 people.	hotel-area: east hotel-internet: yes hotel-book people: 24
Example 4 Dontcare	[sys]: No. I am sorry. there isn’t. something else perhaps? [cuda]:How about an art gallery? I don't care what area it is in.	attraction-type: art gallery attraction-area: dontcare
Example 5 Confirm Coref Domain	[sys]: Okay , we have the cambridge university botanic gardens in the centre of town . Will that work for you ? [cuda]:Yes, I need a taxi to get me to itsuki restaurant by the time of my reservation.	attraction-area: centre attraction-name: cambridge university botanic gardens taxi-destination: itsuki restaurant taxi-arriveby: 15:45

Figure 4: The CUDA-generated examples. The red tags indicate the strategies implemented by CUDA.