CoCo: Controllable Counterfactuals for Evaluating Dialogue State Trackers

Anonymous authors
Paper under double-blind review

Abstract

Dialogue state trackers (DST) have made significant progress on benchmark datasets but their generalization capability to novel and realistic scenarios beyond the held-out conversations is less understood. We propose controllable counterfactuals (CoCo) to bridge this gap and evaluate DST models on novel scenarios, i.e., would the system successfully tackle the request if the user responded differently but still consistent with the dialogue flow? CoCo leverages turn-level belief states as counterfactual conditionals to produce novel conversation scenarios in two steps: (i) *counterfactual goal* generation at turn-level by dropping and adding slots followed by replacing slot values, conditioned on which (ii) *counterfactual conversation* generation that is consistent with the dialogue flow. Evaluating state-of-the-art DST models on MultiWOZ dataset with CoCo-generated counterfactuals results in a significant performance drop of up to 30.8% (from 49.4% to 18.6%) in absolute joint goal accuracy while widely used techniques like paraphrasing only affect the accuracy by at most 2%. The human evaluation shows that CoCo-generated conversations perfectly reflect the underlying user goal with more than 95% accuracy and are as human-like as the original conversations, which further strengthens its reliability and potential to be adopted as part of the robustness evaluation of DST models.

1 Introduction

Dialogue state tracking (DST) is a backbone of pipeline-based task-oriented dialogue (TOD) systems (Young et al., 2013). At each turn of the dialogue, DST component is responsible for estimating the user's goal represented as a set of slot-value pairs (e.g., (*area*, center), (*food*, British)). The output of the DST module is treated as the summary of the user's goal so far in the dialogue and directly consumed by the subsequent dialogue policy component to determine the system's next action and response. Hence, the accuracy of the DST module is critical to prevent the downstream error propagation (Liu and Lane, 2018) affecting the success of the whole system.

With the advent of representation learning in NLP (Pennington et al., 2014; Devlin et al., 2019; Radford et al., 2019), the joint accuracy of DST models has increased from 15.57% (in 2018) to 55.72% (in 2020). While measuring the progress on held-out accuracy is often useful, it is not sufficient; practitioners consistently overestimate their model's generalization (Ribeiro et al., 2020; Patel et al., 2008) since test data is usually collected in the same way as training data as presented in Table 1. The development and test sets have similar distributions, causing potential overfitting of DST models and practitioners to overestimate their model's performance. Inspired by this phenomenon, our primary goal is to address and provide further insights to the following question for state-of-the-art DST models: *how well do these models generalize to novel but realistic scenarios that are simply not captured well enough by the held-out evaluation set?*

Most prior work (Iyyer et al., 2018; Jin et al., 2019) focus on adversarial example generation for robustness evaluation. They often rely on perturbations made directly on test examples in the held-out set and assume direct access to the gradient or outputs of evaluated models. Adversarial examples generated by these methods are often unnatural and/or obtained to deliberately hurt target models. It's very important to emphasize here that both our primary goal and approach significantly differ from the previous line of work: (i) Our goal is to evaluate DST models beyond held-out accuracy, (ii) We leverage turn-level structured meaning representation (belief state) along with dialogue his-

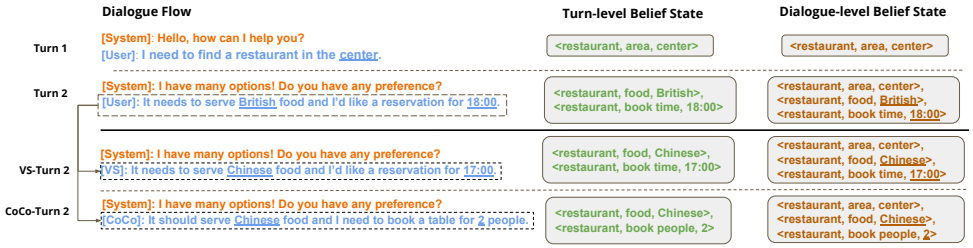

Figure 1: Comparisons between *Value Substitution* (VS) and CoCo. The upper left is a dialog example between *user* and *system* with its turn-level and dialogue-level belief states on the upper right. The lower left are valid user utterance variations generated by VS and CoCo with their corresponding belief states on the right.

slot name	data	area	book day	book time	food	name	price range
book people	train	1.9	38.8	39.2	2.1	16.4	1.5
	dev	1.9	38.9	38.9	1.9	16.3	2.2
	test	2.7	36.9	37.7	1.6	18.7	2.4

Table 1: *Restaurant-book people* co-occurrence distribution (%) with other slots at *restaurant* domain within the same user utterance on MultiWoZ 2.1.

tory as a conditional to generate a user response without relying on the original user utterance, (iii) Our approach is completely model-agnostic assuming no access to evaluated DST models, (iv) Perhaps most importantly, we aim to produce novel but totally realistic and meaningful conversation scenarios rather than intentionally adversarial ones.

We propose *controllable counterfactuals* (CoCo) as a principled, model-agnostic approach to generate novel scenarios beyond the held-out conversations. Our approach is inspired by the combination of two natural questions: how would DST systems react to (1) unseen slot values and (2) rare but realistic slot combinations? CoCo first encapsulates these two aspects under a unified concept called *counterfactual goal*, which is obtained by a stochastic policy of dropping and/or adding slots to the original turn-level belief state followed by replacing slot values, which is illustrated in Figure 1. In the second step, CoCo conditions on the dialogue history and a *counterfactual goal* to generate a *counterfactual conversation*. We cast the actual utterance generation as a conditional language modeling objective. This formulation allows us to plug-in a pretrained encoder-decoder architecture (Raffel et al., 2020) as the backbone that powers the counterfactual conversation generation. We also propose a strategy to filter utterances that fail to exactly reflect the modified user intent. We consider *value substitution* (VS), a special case of CoCo that only replaces the slot values in the original user utterance with the ones in counterfactual goal. When we use VS as a fall-back strategy for CoCo, we call it CoCo+.

Evaluating three strong DST models (Wu et al., 2019; Heck et al., 2020; Hosseini-Asl et al., 2020) with our proposed controllable counterfactuals generated by CoCo and CoCo+ shows that the performance of each model significantly drops (up to 30.8%) compared to their joint goal accuracy on the original MultiWOZ held-out evaluation set. On the other hand, we find that these models are in fact quite robust to paraphrasing with back-translation, where their performance drop up to 2%. Analyzing the effect of training data augmentation with CoCo+ shows that it consistently improves the robustness of the investigated DST models on counterfactual conversations generated by each of VS, CoCo and CoCo+. More interestingly, the same data augmentation strategy improves the joint goal accuracy of best of these strong DST models by 1.26% on the original MultiWOZ evaluation set as well. Human evaluations indicate that the quality of CoCo-generated counterfactual conversations correctly reflect the underlying user goal with more than 95% accuracy and are found to be quite close to original conversations in terms of their human-like scoring. This further proves the reliability and potential of our proposed approach to be adopted as part of robustness evaluation of DST models. We are planning to publicly release CoCo-generated and human-verified set of examples (MultiWOZ-CoCo) as additional evaluation set to be used by future research.

2 RELATED WORK

Dialogue State Tracking DST has been a core component in current state-of-the-art TOD systems. Traditional approaches usually rely on hand-crafted features or domain-specific lexicon (Henderson et al., 2014; Wen et al., 2017) and require a predefined ontology, making them hard to extend to unseen value. To tackle this issue, different methods are proposed. (Gao et al., 2019) treats DST as a reading comprehension problem and predicts slot values with start and end positions in the dialogue context. (Zhang et al., 2019) proposes DS-DST, a dual-strategy model that predicts values in domains with few possible candidates from classifiers and others from span predictors. Further, (Heck et al., 2020) proposes TripPy, a triple copy strategy model, which not only allows the model to copy values from the context but also from previous turns' predictions and system informs.

An alternative to classification and span prediction is value generation. (Wu et al., 2019) generates slot values with a pointer generator network (See et al., 2017) without relying on fixed vocabularies and spans. (Hosseini-Asl et al., 2020) models DST as a conditional generation problem and directly finetunes GPT2 (Radford et al., 2019) on DST task and achieves state-of-the-art on the MultiWOZ.

Adversarial Example Generation Adversarial example generation has been extensively studied in computer vision (Szegedy et al., 2014; Goodfellow et al., 2015), whose inputs are in the context of continuous space, where adversarial examples can be generated by gradient-based optimization of the original inputs. However, generating adversarial examples in natural language is much more challenging as they are discrete in nature. Papernot et al. (2016) searches adversarial examples in embedding space and then map them back to discrete space. Alzantot et al. (2018) proposes a population-based word replacing method and aims to generate fluent adversarial sentences. These methods often edit the original data greedily assuming that they have access to the model's gradients or outputs and they can query the model many times(Jin et al., 2019). There also exists some works that try to generate adversarial examples in a model-agnostic way. (Iyyer et al., 2018) aims at generating adversarial paraphrase with different syntactic structure of original data, which preserves the same semantic meaning but change the model's prediction. (Jia and Liang, 2017) automatically generates sentences with key word overlappings of questions in SQuAD (Rajpurkar et al., 2016) to distract computer systems without changing the correct answer or misleading humans.

Although different methods have been prosed to evaluate the robustness of NLP models, majority of the prior work in this line focus either on text classification, neural machine translation or reading comprehension problems. Perhaps the most similar existing works with ours are (Einolghozati et al., 2019; Cheng et al., 2019). Einolghozati et al. (2019) focuses on intent classification and slot tagging in TOD while Cheng et al. (2019) targets at competitive negotiation dialogues (Lewis et al., 2017) without the need of DST. In this work, however, we focus on DST task in TOD and evaluate our methods on the widely used benchmark, MultiWoZ. To the best of our knowledge, ours is the first work to systematically evaluate the robustness of DST models.

3 BACKGROUND

Multi-domain DST task description Let $X_t = \{(U_1^{\text{sys}}, U_1^{\text{usr}}), ..., (U_t^{\text{sys}}, U_t^{\text{usr}})\}$ denote a sequence of turns of a dialogue until the t-th rurn, where U_i^{sys} and U_i^{usr} $(1 \leq i \leq t)$ denote system and user utterance at the i-th turn, respectively. In multi-domain DST, each turn $(U_i^{\text{sys}}, U_i^{\text{usr}})$ talks about a specific domain (e.g., "hotel"), and a certain number of slots (e.g., "price range") in that domain. We denote all N possible domain-slot pairs as $S = \{S_1, ...S_N\}$. The task is to track the value for each S_j $(1 \leq j \leq N)$ over X_t (e.g., "hotel-price range", "cheap"). There exist two level of labels, namely dialogue-level belief state B_t and turn-level belief state L_t; B_t tracks accumulated S from the first turn to the t-th turn, while L_t tracks S specific to the t-th turn. For example, as shown in the upper part of Fig. 1, when the dialogue flow comes to the second turn, L_2 is {("restaurant-food", "British"), ("restaurant-book time", "18:00")}, while B_2 is {("restaurant-area", "center"), ("restaurant-food", "British"), ("restaurant-book time", "18:00")}.

Problem definition Given a tuple $< X_t, L_t, B_t >$, we want to generate a valid natural user utterance $U_t^{\text{nat_usr}}$ to constitute $\hat{X}_t = \{(U_1^{\text{sys}}, U_1^{\text{usr}}), ..., (U_t^{\text{sys}}, U_t^{\text{nat_usr}})\}$, such that $U_t^{\text{nat_usr}}$ is natural to humans and preserves the coherence of $\hat{L}_t$ and $\hat{B}_t$. We then use $< \hat{X}_t, \hat{L}_t, \hat{B}_t >$ to evaluate the performance of DST models. An example is shown in the lower part of Fig. 1. U_2^{usr} is replaced with the two

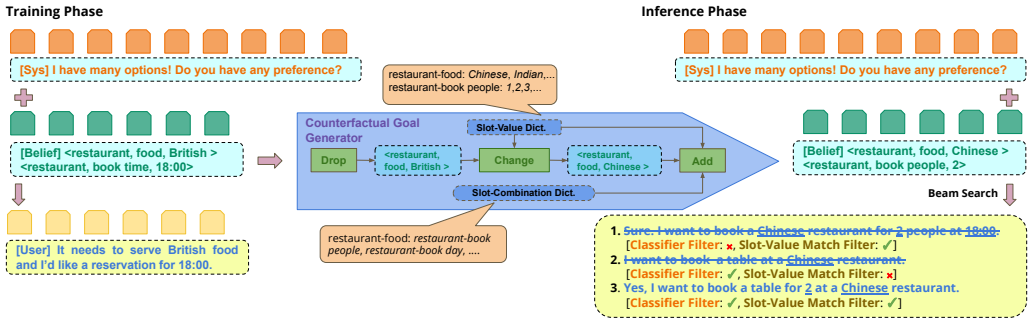

Figure 2: The overall pipeline of CoCo.

alternative utterances that read naturally and preserve the coherence with the previous dialog flow; hence, we can use them to evaluate the DST models on more diverse data.

Paraphrase baseline: back translation A natural way to generate $U_t^{\text{nat_usr}}$ is to paraphrase original utterance, U_t^{usr}. Translation between two languages has been leveraged as a way to extract paraphrases (Ganitkevitch et al., 2013), and back translation (BT) is a widely used method by round-trip translation with neural machine translation (NMT) models (Yu et al., 2018). We use publicly available pretrained *English*→*German* ($\log(g|e)$) and *German*→*English* ($\log(e|g)$) NMT models.[1] We translate U_t^{usr} from English to German with a beam size K, and then translate each of the K hypotheses back to English with the beam size K. Consequently, we generate K^2 paraphrase candidates of $U_t^{\text{nat_usr}}$ and then rank them according to their round-trip confidence score $\log(g|e) + \log(e|g)$. As paraphrases are expected to preserve the meaning of U_t^{usr}, we set $\hat{L}_t = L_t$ and $\hat{B}_t = B_t$.

4 CoCo

The overview of CoCo is illustrated in Fig. 2. We first train a conditional generation model $p(U_t^{\text{usr}}|U_t^{\text{sys}}, L_t)$, and we modify L_t into possible $\hat{L}_t$ by counterfactual goal generator. Given $\hat{L}_t$ and U_t^{sys}, we can sample $U_t^{\text{nat_usr}} \sim p(U_t^{\text{nat_usr}}|U_t^{\text{sys}}, \hat{L}_t)$ with beam search and apply filtering.

4.1 Value Substitution

A robust DST model should correctly reflect value changes in user utterances while tracking user's goals. However, slot-value combinations in a dataset are limited, and to evaluate DST models with more diverse patterns, we propose a Value Substitution (VS) method to generate $U_t^{\text{nat_usr}}$, which is illustrated in Fig 1. At the second turn, as "British" and "18:00" both appear in U_2^{usr} and L_2, we can replace them with "Chinese" and "17:00", respectively, without interrupting the naturalness of the dialogue flow. We sample these new values from a predefined value set for each domain-slot. We can then update substituted values in L_2 and B_2, and get their updated belief states $\hat{L}_2$ and $\hat{B}_2$.

4.2 Controllable Counterfactual Generation

Can we control users requests (represented with slot-value pairs) and generate more diverse types of $U_t^{\text{nat_usr}}$? That is, given U_t^{sys} and $\hat{L}_t$, can we generate $U_t^{\text{nat_usr}}$ such that $(U_t^{\text{sys}}, U_t^{\text{nat_usr}})$ within $\hat{X}_t$ exactly expresses the intents in $\hat{L}_t$? Neither of the VS and BT can achieve this; the VS method can only substitute values for a fixed set of slots, and the BT method is expected to preserve the same meaning. We propose to tackle this problem by a conditional generation model. We convert every dialog turn into a tuple $(U_t^{\text{sys}}, L_t, U_t^{\text{usr}})$ and train a model approximating $p(U_t^{\text{usr}}|U_t^{\text{sys}}, L_t)$. We can instantiate p with any conditional generation models, such as GPT2 and T5 (Raffel et al., 2019).

Once we train such a model on these tuples, we modify L_t into possibly arbitrary $\hat{L}_t$ in the counterfactual goal generator. An example of how the counterfactual goal generator works is shown in

[1] https://pytorch.org/hub/pytorch_fairseq_translation

the middle of Fig. 2. The counterfactual goal generator has three components, namely operation, slot-value dictionary and slot-combination dictionary.

Operation decides to apply which combination of the following three meta-operations, namely *drop*, *change* and *add* on L_t. "Drop" is used to remove a value from a non-empty slot in L_t. "Change" borrows the same operation in VS, to substitute a value. "Add" allows us to add a new domain-slot value into L_t, giving us the power of generate valid but more complicated $U_t^{\text{nat_usr}}$.

Slot-Value Dictionary has a pre-defined value set S_j^{val} for each S_j. Once *change* and/or *add* meta-operation is activated for S_j, counterfactual goal generator will randomly sample a value from S_j^{val}.

Slot-Combination Dictionary has a predefined domain-slot set S_j^{add} for each S_j. When *add* meta-operation is activated, counterfactual goal generator will sample a domain-slot from the intersection among all S_j^{add}, where S_j has non-empty values within L_t. Once a new domains-lot is sampled, its value will then be sampled from its corresponding value set as defined in *slot-value dictionary*.

Given L_t, the counterfactual goal generator first takes L_t as its input, and sequentially applies *drop*, *change* and *add* to output $\hat{L}_t$. We then use a rule-based method to get the dialogue-level belief state of $\hat{X}_t$ following Chao and Lane (2019). Given B_{t-1} and $\hat{L}_t$, we update the domain-slot in B_{t-1}, if its value in $\hat{L}_t$ is not *none*. Otherwise, we keep its value as it is in B_{t-1}. After the update, we get the $\hat{B}_t$ as the dialogue-level label of $\hat{X}_t$.

4.3 Filtering

We have presented methods to generate $U_t^{\text{nat_usr}}$, but there is not guarantee that it satisfies our requirements. Filtering is thus an important step to filter out obviously unqualified candidates which do not exactly express user's goals in $\hat{L}_t$. We proposes two filtering methods, namely slot-value match filter and classifier filter, to alleviate this issue.

Slot-Value Match Filter Although BT can preserve the meaning of U_t^{usr}, name entities (e.g. name of restaurants), which are values for some S_j in $\hat{L}_t$, may change during paraphrase generation process. We also noticed that in CoCo, when $\hat{L}_t$ contains too many (*domain-slot*, *value*) pairs, some beams generated by beam search may ignore some values in $\hat{L}_t$. We propose to utilize *slot-value match filter* to filter out these unqualified $U_t^{\text{usr_nat}}$ candidates. Specifically, we choose a subset of values in $\hat{L}_t$ with predefined value selection rules to filter out candidates that doesn't contain all values in the subset. An example of how slot-value match filter works is demonstrated in the lower right of Fig. 2. In this example, beam 2: *I want to book a table at a Chinese restaurant.* doesn't include "2", the value of *restaurant-book people* in $\hat{L}_t$, making slot-value match filter filter it out.

Classifier Filter As shown in Table 1, the slot "restaurant-book people" frequently appears together with "restaurant-book time" in the training data of our model p. During inference phase of $p(U_t^{\text{nat_usr}}|U_t^{\text{sys}}, \hat{L}_t)$, although $\hat{L}_t$ may not include "restaurant-book time," p may still generate $U_t^{\text{nat_usr}}$ including its value. An example of this problem is shown in the right part of Fig.2. In this example, the first hypothesis in the beam search, "Sure. I want to book a Chinese restaurant for 2 people at 18:00" includes a value "18:00" (for "restaurant-book time"), which is not included in $\hat{L}_t$. *Slot-Value Match Filter* won't be able to filter it out. Therefore, we propose to use a N-way multi-label classifier to filter such candidates. The classifier takes $\hat{X}_t$ as input and it needs to predict if S_i appears at t-th turn or not. We only filter out $U_t^{\text{usr_nat}}$ candidates when the model predicts S_i appearing in them while S_i is not in $\hat{L}_t$. In the aforementioned example, the classifier predict *restaurant-book time* appearing in it while the slot is not in $\hat{L}_t$, the classifier will filter it out.

5 Experiments

5.1 Experimental Setup

We consider three strong multi-domain DST models to evaluate the effect of CoCo-generated counterfactual conversations in several scenarios. TRADE (Wu et al., 2019) builds upon pointer generator network and contains a slot gate module for slots classification and a state generator module to gener-

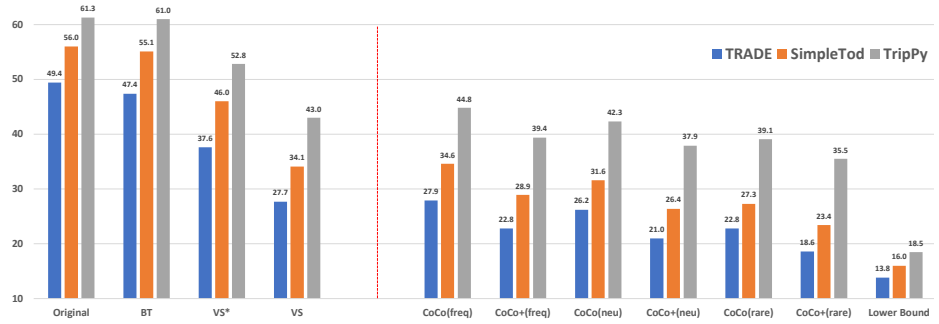

Figure 3: Joint goal accuracy (%) across different methods. "Original" refers to the results on the original held-out test set. * denotes results obtained from in-domain unseen slot-value dictionary (*I*). VS, CoCo and CoCo+ results use out-of-domain slot-value dictionary (*O*). For brevity, we omit CoCo and CoCo+ results using in-domain slot-value dictionary. See Appendix E for the full results. *freq*, *neu*, and *rare* indicate which slot-combination dictionary is used. Lower bound refers to accuracy on turns with empty turn-level belief state over whole held-out test set.

ate states. TripPy (Heck et al., 2020) introduces a classification gate and a triple copy module. The triple copy module allows the model to copy values from the conversation context, previous turns' predictions and system informs. The classification gate will decide which copy mechanism will be activated. SimpleTod (Hosseini-Asl et al., 2020) recasts multi-domain DST as a causal language modeling over the sequences obtained by concatenation of conversation history and dialogue-level belief state. It fine-tunes a GPT2 to model these sequences, which, in the inference, can directly decode the belief state conditioned on the conversation history.

Evaluation. We train each of these three models following their publicly released implementations on the standard train/dev/test split of MultiWOZ 2.1 (Eric et al., 2019) from scratch. We use the joint goal accuracy to evaluate the performance of DST models. It measures the model accuracy by comparing the predicted belief state with the ground-truth, where the prediction is marked correct if and only if the set of (*domain-slot*, value) pairs in the model output exactly matches the oracle set.

Slot-Value Dictionary We carefully design two sets of slot-value dictionaries to capture the effect of unseen slot values from two perspectives, namely *in-domain* (*I*) and *out-of-domain* (*O*). *I* is a dictionary that maps each slot to a set of values that appear in MultiWOZ test set, but not in the training set [2]. On the other hand, we construct *O* using external values (e.g., hotel names from Wikipedia) that fall completely outside of the MultiWOZ data for the slots (e.g., *hotel-name*, *restaurant-name*, etc.). Otherwise, we follow a similar fall-back strategy for slots (e.g., *hotel-internet*) with no possible external values beyond the ones (e.g., "yes" and "no") in the original data.

Slot-Combination Dictionary As illustrated in Table 1, held-out evaluation set follows almost the same slot co-occurrence distribution with training data. This makes it difficult to estimate how well DST models would generalize on the valid conversation scenarios that just do not obey the same distribution. CoCo's flexibility at generating a conversation for an arbitrary turn-level belief state naturally allows us to seek an answer to this question. To this end, we design three slot combination dictionaries, namely *freq*, *neu* and *rare*. A slot combination dictionary directly controls how different slots can be combined while generating counterfactual goals. As suggested by their names, *freq* contains frequently co-occurring slot combinations (e.g., *book people* is combined only with *book day* and *book time* slots), while *rare* is the opposite of *freq* grouping rarely co-occurring slots together, and *neu* is more neutral allowing any meaningful combination within the same domain [3]

5.2 Main Results

Before we begin reporting our results, it is important to note that several different post-processing strategies (e.g., output cleaning, employing semantic dictionary mapping) are used by different DST

[2] When this set is empty for a slot (e.g. *hotel-area*), we use the set of all possible values (e.g., center, west, east, south, north) for this slot from training data. Please see Appendix D for further details.

[3] Please see Appendix C for further details.

models. To make a fair comparison across different models, we follow the same post-processing strategy employed by SIMPLETOD evaluation script for TRADE and TRIPPY as well. We summarize our main results in Figure 3. While all three DST models are quite robust to back-translation (BT), their performance significantly drop on counterfactual conversations generated by each of COCO+, COCO, and VS compared to MultiWOZ held-out set accuracy (original).

Unseen Slot-Value Generalization. We analyze the effect of unseen slot values for the two dictionaries (*I* and *O*) introduced in the previous section compared to the original set of slot values that have large overlap with the training data. Results presented on the left part of Figure 3 show that the performance of DST models significantly drops (8%-9%) compared to original accuracy even on the simple counterfactuals generated by VS strategy using in-domain unseen slot-value dictionary (I). Furthermore, using out-of-domain slot-value dictionary (O) results in about 10% additional drop in accuracy consistently across the three models. Consistent and similar drop in accuracy suggests that TRADE, SIMPLETOD, and TRIPPY are almost equally susceptible to unseen slot values.

Generalization to Novel Scenarios. The right section of Figure 3 presents the main results in our effort to answer the central question we posed at the beginning of this paper. Based on these results, we see that state-of-the-art DST models are having a serious difficulty generalizing to novel scenarios generated by both COCO and COCO+ using three different slot combination strategies. The generalization difficulty become even more serious on counterfactuals generated by COCO+. As expected, the performance drop consistently increases as we start combining less and less frequently co-occurring slots (ranging from *freq* to *rare*) while generating our counterfactual goals. In particular, COCO+(rare) counterfactuals drops the accuracy of TRADE from 49.4% to 18.6%, pushing its performance very close to its lower bound [4] of 13.8%. Even the performance of the most robust model (TRIPPY) among the three drops by up to 25.8%, concluding that held-out accuracy for state-of-the-art DST models may not sufficiently reflect their generalization capabilities.

Transferability Across Models. As highlighted before, a significant difference and advantage of our proposed approach lies in its model-agnostic nature, making it immediately applicable for evaluation of any DST model. As can be inferred from Figure 3, the effect of COCO-generated counterfactuals on the joint goal accuracy is quite consistent across all three DST models. This result empirically proves the transferability of COCO, strengthening its reliability and applicability to be generally employed as a robustness evaluation of DST models by the future research.

5.3 HUMAN EVALUATION

Table 2: Human evaluation.

	Human likeliness	Correctness
Human	87%	85%
COCO(ori)	90%	91%
COCO(freq)	90%	99%
COCO(neu)	79%	98%
COCO(rare)	82%	96%

We next examine the quality of our generated data from two perspectives: "human likeliness" and "turn-level belief state correctness." The human likeliness evaluates whether a user utterance is fluent and consistent with its dialog context. The turn-level belief state correctness evaluates whether a user utterance covers all the slot-value information in its given turn-level belief state. Both metrics are based on binary evaluation. We randomly sample 100 turns in the original test data and their corresponding CoCo-generated ones. For the COCO-generated data, we have two different settings to examine its quality. The first is to use its original turn-level belief state to generate user utterance. We ask three individuals with proficient English and advanced NLP background to conduct the evaluation for original human response both in MultiWoZ and CoCo-generated counterpart. We use majority voting to determine the final human likeness and turn-level belief state correctness.

Table 2 shows the results. We can see that the human evaluators could not distinguish between the MultiWOZ's human utterance and our generated utterance. Furthermore, COCO(ori) generated slightly more "correct" responses than the original utterances in MultiWoZ 2.1. A presumable reason is that annotation errors exist in MultiWoZ 2.1, while our COCO are trained on recently released cleaner MultiWoZ 2.2, making generated data have higher quality. The second setting is to evaluate COCO(freq)-, COCO(neu)- and COCO(rare)-generated data, as they hurt the DST models' accuracy significantly as show in Fig. 3, and we need to verify the quality of the generated utterances. All three variants of the COCO-generated conversations consistently outperform human response in

[4] As we only touch turns with non-empty turn belief state, lower bound refers to percentage of correct predictions on turns with empty turn-level belief state over held-out test set.

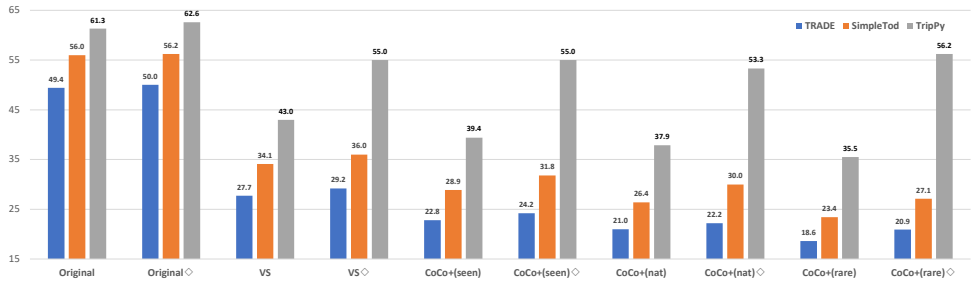

Figure 4: Comparison of retrained DST models (indicated by ◇) on CoCo+(rare)-augmented training data with their counterparts trained on original MultiWOZ train split. Original, VS and various CoCo+ versions share the same meaning as in Figure 3.

term of the turn-level belief state correctness. Although CoCo(neu) and CoCo(rare) are slightly less human-like than the original human response, CoCo(freq)-generated utterances are found to be difficult to distinguish from the original human utterances. These results demonstrate the effectiveness of our proposed approach in generating not only high-fidelity but also human-like user utterances, proving its potential to be adopted as part of robustness evaluation of DST models.

5.4 Analysis of CoCo+ as Data Augmentation Defense

So far, we have focused on the generalization capability of DST models on CoCo-generated conversations using different slot-value and slot-combination dictionaries. We have observed that all three DST models are consistently most susceptible to conversations generated by CoCo+(rare) strategy. Instead, we now seek to answer the following question: *Would using conversations generated by* CoCo+*(rare) to augment the training data help these DST models in better generalizing to unseen slot values and/or novel scenarios?* Towards exploring this direction in a principled way, we design a new slot value dictionary (*train-O*) similar to out-of-domain unseen slot-value dictionary. For a completely fair comparison, we make sure that the slot values in *train-O* (please refer to Appendix D for the complete dictionary) do not overlap with the one (*O*) used for generating test conversations.

We first retrain each DST model on the MultiWOZ training split augmented with CoCo+(rare)-generated conversations using *train-O* slot-value dictionary. Retrained DST models are then evaluated on original test set as well as on the counterfactual test sets generated by VS and various versions of CoCo+. Results presented in Figure 4 shows that retraining on the CoCo+(rare)-augmented training data improves the robustness of all three DST models across the board. Most notably, this data augmentation strategy rebounds the performance of TripPy on CoCo+(rare)-generated test set from 35.5% to 56.2%, significantly closing the gap with its performance (61.3%) on the original held-out test set. We also observe that retrained DST models obtains an improved joint goal accuracy on the original MultiWOZ test set compared to their counterparts trained only on the original MultiWOZ train split, further validating the quality of CoCo-generated conversations. Finally, we would like to highlight that retrained TripPy model achieves 62.6% joint goal accuracy, improving the previous state-of-the-art by 1.3%. We leave the exploration of how to fully harness CoCo as a data augmentation approach as future work.

6 Conclusion

We propose a principled, model-agnostic approach (CoCo) to evaluate dialogue state trackers beyond the held-out evaluation set. The significant performance drop of these state-of-the-art DST models when evaluated on the CoCo-generated counterfactual conversations using different unseen slot-value and slot-combination dictionaries implies that they are vulnerable to (1) unseen slot values and (2) novel scenarios. The human evaluation demonstrates that CoCo-generated conversations have high-fidelity and are human-like. When explored as a data augmentation method, CoCo improves the performance of state-of-the-art DST models not only on the CoCo-generated evaluation set, but also on the original test set. This further proves the reliability and potential of our proposed approach to be adopted as part of the robustness evaluation of DST models.

REFERENCES

M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1316. URL https://www.aclweb.org/anthology/D18-1316.

G. Chao and I. Lane. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. In *INTERSPEECH*, 2019.

M. Cheng, W. Wei, and C.-J. Hsieh. Evaluating and enhancing the robustness of dialogue systems: A case study on a negotiation agent. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3325–3335, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1336. URL https://www.aclweb.org/anthology/N19-1336.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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*, 2019.

A. Einolghozati, S. Gupta, M. Mohit, and R. Shah. Improving robustness of task oriented dialog systems. *ArXiv*, abs/1911.05153, 2019.

M. Eric, R. Goel, S. Paul, A. Kumar, A. Sethi, A. K. Goyal, P. Ku, S. Agarwal, S. Gao, and D. Z. Hakkani-Tür. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *ArXiv*, abs/1907.01669, 2019.

J. Ganitkevitch, B. Van Durme, and C. Callison-Burch. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1092.

S. Gao, A. Sethi, S. Agarwal, T. Chung, and D. Z. Hakkani-Tür. Dialog state tracking: A neural reading comprehension approach. *ArXiv*, abs/1908.01946, 2019.

I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

M. Heck, C. van Niekerk, N. Lubis, C. Geishauser, H.-C. Lin, M. Moresi, and M. Gasic. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting, July 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.sigdial-1.4.

M. Henderson, B. Thomson, and S. Young. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, Philadelphia, PA, U.S.A., June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-4340. URL https://www.aclweb.org/anthology/W14-4340.

E. Hosseini-Asl, B. McCann, C.-S. Wu, S. Yavuz, and R. Socher. A simple language model for task-oriented dialogue, 2020.

M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1170. URL https://www.aclweb.org/anthology/N18-1170.

R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL https://www.aclweb.org/anthology/D17-1215.

D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits. Is bert really robust? natural language attack on text classification and entailment. *ArXiv*, abs/1907.11932, 2019.

M. Lewis, D. Yarats, Y. Dauphin, D. Parikh, and D. Batra. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1259. URL https://www.aclweb.org/anthology/D17-1259.

B. Liu and I. Lane. End-to-end learning of task-oriented dialogs. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

N. Papernot, P. McDaniel, A. Swami, and R. E. Harang. Crafting adversarial input sequences for recurrent neural networks. *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 49–54, 2016.

K. Patel, J. Fogarty, J. A. Landay, and B. Harrison. Investigating statistical machine learning as a tool for software development. In *CHI*, 2008.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

A. Radford, J. Wu, R. Child, L. David, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683, 2019.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100, 000+ questions for machine comprehension of text. *ArXiv*, abs/1606.05250, 2016.

M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://www.aclweb.org/anthology/P17-1099.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gašić, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain, Apr. 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1042.

C.-S. Wu, A. Madotto, E. Hosseini-Asl, C. Xiong, R. Socher, and P. Fung. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2019.

S. Young, M. Gašić, B. Thomson, and J. D. Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *6th International Conference on Learning Representations (ICLR)*, 2018.

J. Zhang, K. Hashimoto, C.-S. Wu, Y. Wan, P. S. Yu, R. Socher, and C. Xiong. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *ArXiv*, abs/1910.03544, 2019.

APPENDIX

A SLOT-LEVEL ANALYSIS

Closer Look at the Effect of COCO+(rare) on TRIPPY. In Figure 5, we take a closer look at the robustness of TRIPPY through slot-level analysis across three major scenarios. Comparison of *blue* and *orange* lines reveals that counterfactuals generated by COCO+(rare) consistently drops the performance of TRIPPY model (trained on the original MultiWOZ train split) across all the slots, significantly hurting the accuracy of all the slots in *train* domain along with *book day* slot for *hotel* and *restaurant* domains. On the other hand, comparing *green* and *orange* lines clearly demonstrates the effectiveness of COCO+(rare) as a data augmentation defense (see Section 5.4 for further details), assisting TRIPPY in recovering from most of the errors it made on COCO+(rare) evaluation set. In fact, it rebounds the joint goal accuracy of TRIPPY from 35.5% to 56.2% as presented more quantitatively in Figure 4.

Figure 5: Slot-level accuracy analysis of TRIPPY. "Ori-TripPy-Clean" (blue) and "Ori-TripPy-CoCo+(rare)" (orange) denote TRIPPY (trained on original MultiWOZ training data) when evaluated against original test set and COCO+(rare) generated test set, respectively. "Aug-TripPy-CoCo+(rare)" (green) indicates slot-level accuracy of TRIPPY when evaluated against test set generated by COCO+(rare)

B Ablation Study on Operations

In Table 3, we present ablation results on the meta operations that are used to generate counterfactual goals. The result in the first row corresponds to the DST accuracy of models on evaluation set generated by CoCo including all of the meta operations (i.e., drop, change, add) along with the classifier filtering. Each row analyzes the effect of the corresponding meta operation or classifier by removing it from the full model. We observe from Table 3 that "add" operation has generally the highest impact on the performance of all the three models. This may indicate that the investigated DST models are vulnerable against user utterances including more slots.

Table 3: Ablation study on the meta operations and classifier based filtering.

CoCo	TRADE	SIMPLETOD	TRIPPY
Full	26.17	31.60	42.33
-Drop	25.73	31.05	42.14
-Change	30.39	35.95	50.38
-Add	34.13	40.95	48.33
-Classifier	25.30	30.50	41.33

C SLOT-COMBINATION DICTIONARY

Please find the different slot-combination dictionaries introduced in the main paper below.

domain-slot	*freq*
”hotel-internet”	[”hotel-area”,”hotel-parking”,”hotel-pricerange”,”hotel-stars”,”hotel-type”]
”hotel-type”	[”hotel-area”,”hotel-internet”,”hotel-parking”,”hotel-pricerange”,”hotel-stars”]
”hotel-parking”	[”hotel-area”,”hotel-internet”,”hotel-pricerange”,”hotel-stars”,”hotel-type”]
”hotel-pricerange”	[”hotel-area”,”hotel-internet”,”hotel-parking”,”hotel-stars”,”hotel-type”]
”hotel-book day”	[”hotel-book people”,”hotel-book stay”]
”hotel-book people”:	[”hotel-book day”,”hotel-book stay”]
”hotel-book stay”	[”hotel-book day”,”hotel-book people”]
”hotel-stars”	[”hotel-area”,”hotel-internet”,”hotel-parking”,”hotel-pricerange”,”hotel-type”]
”hotel-area”	[”hotel-internet”,”hotel-parking”,”hotel-pricerange”,”hotel-stars”,”hotel-type”]
”hotel-name”	[”hotel-book day”,”hotel-book people”,”hotel-book stay”]
”restaurant-area”	[”restaurant-food”,”restaurant-pricerange”]
”restaurant-food”	[”restaurant-area”,”restaurant-pricerange”]
”restaurant-pricerange”	[”restaurant-area”,”restaurant-food”]
”restaurant-name”	[”restaurant-book day”,”restaurant-book people”,”restaurant-book time”]
”restaurant-book day”	[”restaurant-book people”,”restaurant-book time”]
”restaurant-book people”	[”restaurant-book day”,”restaurant-book time”]
”restaurant-book time”	[”restaurant-book day”,”restaurant-book people”]
”taxi-arriveby”	[”taxi-leaveat”,”train-book people”]
”taxi-leaveat”	[”taxi-arriveby”,”train-book people”]
”taxi-departure”	[”taxi-destination”,”taxi-leaveat”,”taxi-arriveby”]
”taxi-destination”	[”taxi-departure”,”taxi-arriveby”,”taxi-leaveat”]
”train-arriveby”	[”train-day”,”train-leaveat”,”train-book people”]
”train-departure”	[”train-arriveby”,”train-leaveat”,”train-destination”,”train-day”,”train-book people”]
”train-destination”	[”train-arriveby”,”train-leaveat”,”train-departure”,”train-day”,”train-book people”]
”train-day”	[”train-arriveby”,”train-leaveat”,”train-book people”]
”train-leaveat”	[”train-day”]
”train-book people”	[]
”attraction-name”	[]
”attraction-area”	[”attraction-type”]
”attraction-type”	[”attraction-area”]

Table 4: Slot-combination dictionary for *freq* case.

slot-name	*neu*
'hotel-internet'	['hotel-book day','hotel-name','hotel-book stay','hotel-pricerange', 'hotel-stars','hotel-area','hotel-book people','hotel-type','hotel-parking']
'hotel-area'	['hotel-book day','hotel-name','hotel-book stay','hotel-pricerange', 'hotel-stars','hotel-book people','hotel-internet','hotel-type','hotel-parking']
'hotel-parking'	['hotel-book day','hotel-name','hotel-book stay','hotel-pricerange','hotel-stars', 'hotel-area','hotel-book people','hotel-internet','hotel-type']
'hotel-pricerange'	['hotel-book day','hotel-name','hotel-book stay','hotel-stars','hotel-area', 'hotel-book people','hotel-internet','hotel-type','hotel-parking']
'hotel-stars'	['hotel-book day','hotel-name','hotel-book stay','hotel-pricerange','hotel-area', 'hotel-book people','hotel-internet','hotel-type','hotel-parking']
'hotel-type'	['hotel-book day','hotel-book stay','hotel-pricerange','hotel-stars','hotel-area', 'hotel-book people','hotel-internet','hotel-parking']
'hotel-name'	['hotel-book day','hotel-book stay','hotel-pricerange','hotel-stars','hotel-area', 'hotel-book people','hotel-internet','hotel-parking']
'hotel-book day'	['hotel-name','hotel-book stay','hotel-pricerange','hotel-stars','hotel-area', 'hotel-book people','hotel-internet','hotel-type','hotel-parking']
'hotel-book people'	['hotel-book day','hotel-name','hotel-book stay','hotel-pricerange','hotel-stars', 'hotel-area','hotel-internet','hotel-type','hotel-parking']
'hotel-book stay'	['hotel-book day','hotel-name','hotel-pricerange','hotel-stars','hotel-area', 'hotel-book people','hotel-internet','hotel-type','hotel-parking']
'restaurant-area'	['restaurant-book day','restaurant-name','restaurant-food','restaurant-book people', 'restaurant-book time','restaurant-pricerange']
'restaurant-food'	['restaurant-book day','restaurant-book people','restaurant-book time', 'restaurant-area','restaurant-pricerange']
'restaurant-pricerange'	['restaurant-book day','restaurant-name','restaurant-food','restaurant-book people', 'restaurant-book time','restaurant-area']
'restaurant-name'	['restaurant-book day','restaurant-book people','restaurant-book time', 'restaurant-area','restaurant-pricerange']
'restaurant-book day'	['restaurant-name','restaurant-food','restaurant-book people','restaurant-book time', 'restaurant-area','restaurant-pricerange']
'restaurant-book people'	['restaurant-book day','restaurant-name','restaurant-food','restaurant-book time', 'restaurant-area','restaurant-pricerange']
'restaurant-book time'	['restaurant-book day','restaurant-name','restaurant-food','restaurant-book people', 'restaurant-area','restaurant-pricerange']
'taxi-departure'	['taxi-destination', 'taxi-leaveat', 'taxi-arriveby']
'taxi-destination'	['taxi-departure', 'taxi-leaveat', 'taxi-arriveby']
'taxi-leaveat'	['taxi-departure', 'taxi-destination', 'taxi-arriveby']
'taxi-arriveby'	['taxi-departure', 'taxi-destination', 'taxi-leaveat']
'train-arriveby'	['train-book people','train-day','train-leaveat','train-departure','train-destination']
'train-leaveat'	['train-book people','train-arriveby','train-day','train-departure','train-destination']
'train-departure'	['train-book people','train-arriveby','train-day','train-leaveat','train-destination']
'train-destination'	['train-book people','train-arriveby','train-day','train-leaveat','train-departure']
'train-day'	['train-book people','train-arriveby','train-leaveat','train-departure','train-destination']
'train-book people'	['train-arriveby','train-day','train-leaveat','train-departure','train-destination']
'attraction-name'	['attraction-area']
'attraction-area'	['attraction-type']
'attraction-type'	['attraction-area']

Table 5: Slot-combination dictionary for *neu* case.

slot-name	*rare*
'hotel-internet'	['hotel-book people','hotel-book day','hotel-name','hotel-book stay']
'hotel-area':	['hotel-book people','hotel-book day','hotel-name','hotel-book stay']
'hotel-parking'	['hotel-book people','hotel-book day','hotel-name','hotel-book stay']
'hotel-pricerange'	['hotel-book people','hotel-book day','hotel-name','hotel-book stay']
'hotel-stars'	['hotel-book people','hotel-book day','hotel-name','hotel-book stay']
'hotel-type'	['hotel-book people','hotel-book day','hotel-book stay']
'hotel-name'	['hotel-pricerange','hotel-stars','hotel-area','hotel-internet','hotel-parking']
'hotel-book day'	['hotel-name','hotel-pricerange','hotel-stars','hotel-area','hotel-internet', 'hotel-type','hotel-parking']
'hotel-book people'	['hotel-name','hotel-pricerange','hotel-stars','hotel-area','hotel-internet', 'hotel-type','hotel-parking']
'hotel-book stay'	['hotel-name','hotel-pricerange','hotel-stars','hotel-area','hotel-internet', 'hotel-type','hotel-parking']
'restaurant-area'	['restaurant-book day','restaurant-name','restaurant-book time', 'restaurant-book people']
'restaurant-food'	['restaurant-book day','restaurant-book time','restaurant-book people']
'restaurant-pricerange'	['restaurant-book day','restaurant-name','restaurant-book time', 'restaurant-book people']
'restaurant-name'	['restaurant-area','restaurant-pricerange']
'restaurant-book day'	['restaurant-name','restaurant-area','restaurant-food','restaurant-pricerange']
'restaurant-book people'	['restaurant-name','restaurant-area','restaurant-food','restaurant-pricerange']
'restaurant-book time'	['restaurant-name','restaurant-area','restaurant-food','restaurant-pricerange']
'taxi-departure'	[]
'taxi-destination'	[]
'taxi-leaveat'	['taxi-departure', 'taxi-destination']
'taxi-arriveby'	['taxi-departure', 'taxi-destination']
'train-arriveby'	['train-destination', 'train-departure']
'train-leaveat'	['train-destination','train-book people','train-arriveby','train-departure']
'train-departure'	[]
'train-destination'	[]
'train-day'	['train-destination', 'train-departure']
'train-book people'	['train-arriveby','train-departure','train-destination','train-day','train-leaveat']
'attraction-name'	['attraction-area']
'attraction-area'	['attraction-name']
'attraction-type'	[]

Table 6: Slot-combination dictionary for *rare* case.

D SLOT-VALUE DICTIONARY

Please find the different slot-value dictionaries introduced in the main paper below.

slot-name	train-O
”hotel-internet”	[’yes’]
”hotel-type”	[’hotel’, ’guesthouse’]
”hotel-parking”	[’yes’]
”hotel-pricerange”:	[’moderate’, ’cheap’, ’expensive’]
”hotel-book day”	[”march 11th”, ”march 12th”, ”march 13th”, ”march 14th”, ”march 15th”, ”march 16th”, ”march 17th”,”march 18th”, ”march 19th”, ”march 20th”]
”hotel-book people”	[”20”,”21”,”22”,”23”,”24”,”25”,”26”,”27”,”28”,”29”]
”hotel-book stay”	[”20”,”21”,”22”,”23”,”24”,”25”,”26”,”27”,”28”,”29”]
”hotel-area”	[’south’, ’north’, ’west’, ’east’, ’centre’]
”hotel-stars”	[’0’, ’1’, ’2’, ’3’, ’4’, ’5’]
”hotel-name”	[”moody moon”, ”four seasons hotel”, ”knights inn”, ”travelodge”, ”jack summer inn”, ”paradise point resort”]
”restaurant-area”	[’south’, ’north’, ’west’, ’east’, ’centre’]
”restaurant-food”	[’asian fusion’, ’burger’, ’pasta’, ’ramen’, ’taiwanese’]
”restaurant-pricerange”:	[’moderate’, ’cheap’, ’expensive’]
”restaurant-name”	[”buddha bowls”,”pizza my heart”,”pho bistro”, ”sushiya express”,”rockfire grill”,”itsuki restaurant”]
”restaurant-book day”	[”monday”,”tuesday”,”wednesday”,”thursday”,”friday”, ”saturday”,”sunday”]
”restaurant-book people”	[”20”,”21”,”22”,”23”,”24”,”25”,”26”,”27”,”28”,”29”]
”restaurant-book 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”]
”taxi-arriveby”	[”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-leaveat”:	[”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-departure”:	[”moody moon”, ”four seasons hotel”, ”knights inn”, ”travelodge”, ”jack summer inn”, ”paradise point resort”],
”taxi-destination”:	[”buddha bowls”,”pizza my heart”,”pho bistro”, ”sushiya express”,”rockfire grill”,”itsuki restaurant”]
”train-arriveby”:	[”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-leaveat”:	[”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-departure”	[”gilroy”,”san martin”,”morgan hill”,”blossom hill”, ”college park”,”santa clara”,”lawrence”,”sunnyvale”]
”train-destination”	[”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-book people”	[”20”,”21”,”22”,”23”,”24”,”25”,”26”,”27”,”28”,”29”]
”attraction-area”	[’south’, ’north’, ’west’, ’east’, ’centre’]
”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’]

Table 7: Slot value dictionary of train-O.

slot-name	I
”hotel-internet”	[’yes’]
”hotel-type”	[’hotel’, ’guesthouse’]
”hotel-parking”	[’yes’]
”hotel-pricerange”	[’moderate’, ’cheap’, ’expensive’]
”hotel-book day”	[’friday’, ’tuesday’, ’thursday’, ’saturday’, ’monday’, ’sunday’, ’wednesday’]
”hotel-book people”	[’1’, ’2’, ’3’, ’4’,’5’, ’6’, ’7’,’8’]
”hotel-book stay”	[’1’, ’2’, ’3’, ’4’,’5’, ’6’, ’7’,’8’]
”hotel-name”	[’alpha milton’, ’flinches bed and breakfast’, ’express holiday inn by cambridge’, ’wankworth house’, ’alexander b and b’, ’the gonville hotel’]
”hotel-stars”	[’0’, ’1’, ’3’, ’2’, ’4’, ’5’]
”hotel-area”	[’south’, ’east’, ’west’, ’north’, ’centre’]
”restaurant-area”	[’south’, ’east’, ’west’, ’north’, ’centre’]
”restaurant-food”	[’europeon’, ’brazliian’, ’weish’]
”restaurant-pricerange”	[’moderate’, ’cheap’, ’expensive’]
”restaurant-name”:	[’pizza hut in cherry’, ’the nirala’, ’barbakan’, ’the golden house’, ’michaelhouse’, ’bridge’, ’varsity restaurant’,’loch’, ’the peking’, ’charlie’, ’cambridge lodge’, ’maharajah tandoori’]
”restaurant-book day”	[’friday’, ’tuesday’, ’thursday’, ’saturday’, ’monday’, ’sunday’, ’wednesday’]
”restaurant-book people”	[’8’, ’6’, ’7’, ’1’, ’3’, ’2’, ’4’, ’5’]
”restaurant-book time”	[’14:40’, ’19:00’, ’15:15’, ’9:30’, ’7 pm’, ’11 am’, ’8:45’]
”taxi-arriveby”	[’08:30’, ’9:45’]
”taxi-leaveat”	[’7 pm’, ’3:00’]
”taxi-departure”	[’aylesbray lodge’, ’fitzbillies’, ’uno’, ’zizzi cambridge’, ’express by holiday inn’, ’great saint marys church’, ’county folk museum’,’riverboat’, ’bishops stortford’, ’caffee uno’, ’hong house’, ’gandhi’, ’cambridge arts’, ’the hotpot’, ’regency gallery’, ’saint johns chop shop house’] ,
”taxi-destination”	[’ashley’, ’all saints’, ”de luca cucina and bar’s”, ’the lensfield hotel’, ’oak bistro’, ’broxbourne’, ’sleeperz hotel’, ”saint catherine’s college”]
”train-arriveby”	[’4:45 pm’, ’18:35’, ’21:08’, ’19:54’, ’10:08’, ’13:06’, ’15:24’, ’07:08’, ’16:23’, ’8:56’, ’09:01’, ’10:23’, ’10:00 am’, ’16:44’, ’6:15’, ’06:01’, ’8:54’,’21:51’, ’16:07’, ’12:43’, ’20:08’, ’08:23’, ’12:56’, ’17:23’, ’11:32’, ’20:54’, ’20:06’, ’14:24’, ’18:10’, ’20:38’, ’16:06’, ’3:00’, ’22:06’, ’20:20’, ’17:51’,’19:52’, ’7:52’, ’07:44’, ’16:08’],
”train-leaveat”	[’13:36’, ’15:17’, ’14:21’, ’3:15 pm’, ’6:10 am’, ’14:40’, ’5:40’, ’13:40’, ’17:11’, ’13:50’, ’5:11’, ’11:17’, ’5:01’, ’13:24’, ’5:35’, ’07:00’, ’8:08’, ’7:40’, ’11:54’, ’12:06’, ’07:01’, ’18:09’, ’13:17’, ’21:45’, ’06:40’, ’01:44’, ’9:17’, ’20:21’, ’20:40’, ’08:11’, ’07:35’, ’14:19’, ’1 pm’, ’19:17’, ’19:48’, ’19:50’, ’10:36’, ’09:19’, ’19:35’, ’8:06’, ’05:29’, ’17:50’, ’15:16’, ’09:17’, ’7:35’, ’5:29’, ’17:16’, ’14:01’, ’10:21’, ’05:01’, ’15:39’, ’15:01’, ’10:11’, ’08:01’],
”train-departure”:	[’london liverpool street’, ’kings lynn’, ’norwich’, ’birmingham new street’, ’london kings cross’,’broxbourne’]
”train-destination”	[’bishops stortford’, ’cambridge’, ’ely’, ’stansted airport’, ’peterborough’, ’leicester’, ’stevenage’]
”train-day”	[’friday’, ’tuesday’, ’thursday’, ’monday’, ’saturday’, ’sunday’, ’wednesday’]
”train-book people”	[’9’]
”attraction-name”:	[’the cambridge arts theatre’, ’the churchill college’, ’the castle galleries’, ’cambridge’, ”saint catherine’s college”, ’street’, ’corn cambridge exchange’, ’fitzwilliam’, ’cafe jello museum’],
”attraction-area”:	[’south’, ’east’, ’west’, ’north’, ’centre’],
”attraction-type”	[’concerthall’, ’museum’, ’entertainment’, ’college’, ’multiple sports’, ’hiking’, ’architecture’, ’theatre’, ’cinema’, ’swimmingpool’, ’boat’, ’nightclub’, ’park’]

Table 8: Slot-value dictionary for I case.

slot-name	*O*
”hotel-internet”	[’yes’]
”hotel-type”	[’hotel’, ’guesthouse’]
”hotel-parking”	[’yes’]
”hotel-pricerange”	[’moderate’, ’cheap’, ’expensive’]
”hotel-book day”	[”april 11th”, ”april 12th”, ”april 13th”, ”april 14th”, ”april 15th”, ”april 16th”, ”april 17th”,”april 18th”, ”april 19th”, ”april 20th”]
”hotel-book people”	[”30”,”31”,”32”,”33”,”34”,”35”,”36”,”37”,”38”,”39”]
”hotel-book stay”	[”30”,”31”,”32”,”33”,”34”,”35”,”36”,”37”,”38”,”39”]
”hotel-area”	[’south’, ’east’, ’west’, ’north’, ’centre’]
”hotel-stars”	[’0’, ’1’, ’2’, ’3’, ’4’, ’5’]
”hotel-name”	[”white rock hotel”, ”jade bay resort”, ”grand hyatt”, ”hilton garden inn” ,”cottage motel”,”mandarin oriental”],
”restaurant-area”	[’south’, ’east’, ’west’, ’north’, ’centre’]
”restaurant-food”	[”sichuan”, ”fish”, ”noodle”, ”lobster”, ”burrito”, ”dumpling”, ”curry”,”taco”]
”restaurant-pricerange”	[’moderate’, ’cheap’, ’expensive’]
”restaurant-name”:	[”lure fish house”,”black sheep restaurant”,”palapa restaurant”, ”nikka ramen”, ”sun sushi”,”super cucas”]
”restaurant-book day”:	[”monday”,”tuesday”,”wednesday”,”thursday”,”friday”,”saturday”,”sunday”]
”restaurant-book people”	[”30”,”31”,”32”,”33”,”34”,”35”,”36”,”37”,”38”,”39”]
”restaurant-book time”	[”20:02”,”21:07”,”22:12”,”20:17”,”21:22”,”22:27”,”20:32”,”21:37”,”22:42”, ”20:47”,”21:52”,”22:57”,”8:00 pm”,”9:04 pm”,”10:09 pm”,”8:14 pm”, ”9:19 pm”,”10:24 pm”,”8:29 pm”,”9:34 pm”,”10:39 pm”,”8:44 pm”,”9:49 pm”, ”10:54 pm”,”10:00 am”,”10:06 am”,”10:11 am”,”10:16 am”,”10:21 am”,”10:26 am”, ”10:31 am”,”10:36 am”,”10:41 am”,”10:46 am”,”10:51 am”,”10:56 am”],
”taxi-arriveby”:	[”20:02”,”21:07”,”22:12”,”20:17”,”21:22”,”22:27”,”9:34 pm”,”10:39 pm”, ”8:44 pm”,”9:49 pm”,”10:54 pm”, ”10:00 am”,”10:06 am”,”10:11 am”, ”10:16 am”,”10:21 am”,”10:26 am”],
”taxi-leaveat”:	[”21:37”,”22:42”,”20:47”,”21:52”,”22:57”,”8:00 pm”,”9:04 pm”,”10:09 pm”, ”8:14 pm”,”9:19 pm”,”10:24 pm”,”8:29 pm”,”10:31 am”,”10:36 am”, ”10:41 am”, ”10:46 am”,”10:51 am”,”10:56 am”],
”taxi-departure”:	[”lure fish house”,”black sheep restaurant”,”palapa restaurant”, ”nikka ramen”, ”sun sushi”,”super cucas”],
”taxi-destination”:	[”white rock hotel”, ”jade bay resort”, ”grand hyatt”, ”hilton garden inn”, ”cottage motel”,”mandarin oriental”]
”train-departure”	[”northridge”,”camarillo”,”oxnard”,”morepark”,”simi valley”,”chatsworth”, ”van nuys”,”glendale”]
”train-destination”	[”norwalk”,”buena park”,”fullerton”,”santa ana”,”tustin”,”irvine”, ”san clemente”,”oceanside”],
”train-arriveby”:	[”20:02”,”21:07”,”22:12”,”20:17”,”21:22”,”22:27”, ”9:34 pm”,”10:39 pm”, ”8:44 pm”,”9:49 pm”,”10:54 pm”,”10:00 am”,”10:06 am”,”10:11 am”, ”10:16 am”,”10:21 am”,”10:26 am”],
”train-day”:	[”april 11th”, ”april 12th”, ”april 13th”, ”april 14th”, ”april 15th”, ”april 16th”, ”april 17th”,”april 18th”, ”april 19th”, ”april 20th”],
”train-leaveat”:	[”21:37”,”22:42”,”20:47”,”21:52”,”22:57”,”8:00 pm”,”9:04 pm”,”10:09 pm”, ”8:14 pm”,”9:19 pm”,”10:24 pm”,”8:29 pm”,”10:31 am”,”10:36 am”, ”10:41 am”,”10:46 am”,”10:51 am”,”10:56 am”],
”train-book people”:	[”30”,”31”,”32”,”33”,”34”,”35”,”36”,”37”,”38”,”39”]
”attraction-area”:	[’south’, ’east’, ’west’, ’north’, ’centre’]
”attraction-name”:	[”statue of liberty”,”empire state building”,”mount rushmore”, ”brooklyn bridge”,”lincoln memorial”,”times square”],
”attraction-type”:	[”temple”, ”zoo”, ”library”, ”skyscraper”,”monument”]

Table 9: Slot-value dictionary for *O* case.

E FULL FIGURE FOR MAIN RESULT

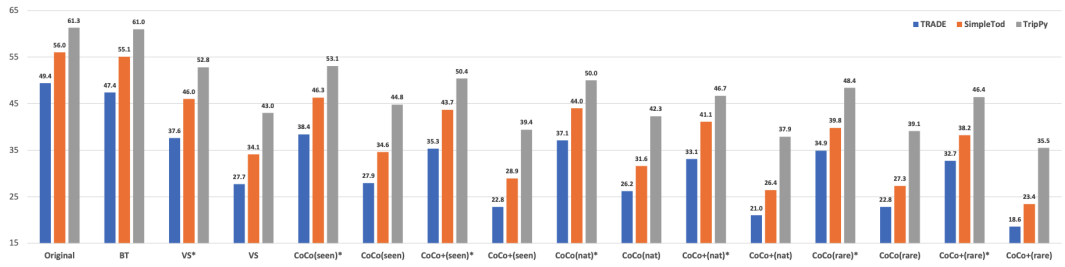

Figure 6: Joint goal accuracy (%) across different methods. "Original" refers to the results on the original held-out test set. * denotes results obtained from in-domain unseen slot-value dictionary (I) while other results use out-of-domain slot-value dictionary (O). *freq*, *neu*, and *rare* indicate which slot-combination dictionary is used.