N-Shot Learning for Augmenting Task-Oriented Dialogue State Tracking

Anonymous ARR submission

Abstract

We introduce an augmentation framework that utilizes belief state annotations to match turns from various dialogues and forms new synthetic dialogues in a bottom-up manner. Unlike other augmentation strategies, it operates with as few as five examples. Our augmentation strategy yields significant improvements when both adapting a DST model to a new domain, and when adapting a language model to the DST task, on evaluations with TRADE and TOD-BERT models. Further analysis shows that our model performs better on seen values during training, and it is also more robust to unseen values even though we do not use any external dataset for augmentation. We conclude that exploiting belief state annotations enhances dialogue augmentation and results improved models in n-shot training scenarios.

1 Introduction

Task-oriented dialogue (TOD) agents are the next-generation user interface and are slated to replace browsing static websites. However, a key bottleneck in fielding such agents practically concerns data availability. As current state-of-the-art methods require ample amounts of annotated data, **data augmentation** is growing in importance (Feng et al., 2021). Most augmentation methods in natural language processing (NLP) target written forms of text — passages, news articles, *etc.* — which operate with word- or sentence-level permutations of the original text data, synthesizing new text (Liu et al., 2020; Wei and Zou, 2019; Yu et al., 2018; Xie et al., 2017; Kobayashi, 2018). These methods do not exploit the structure of conversational data in its entirety. We study augmenting task-oriented dialogues, a specific form of conversational data.

A TOD aims to accomplish a task through exchanges between a user and an agent, accounting for the user's preferences. Within TOD, dialogue state tracking (DST) is a fundamental task, which aims to detect these preferences in a given dialogue. For this task, each pair of utterances in a dialogue is annotated with slot-label and slot-value pairs (*cf.* Figure 1: *train-destination*: "Cambridge") and a belief state. Here, a *belief state* can be equated as an attribute–value store that gives the final values of each slot label (attribute) after an utterance.

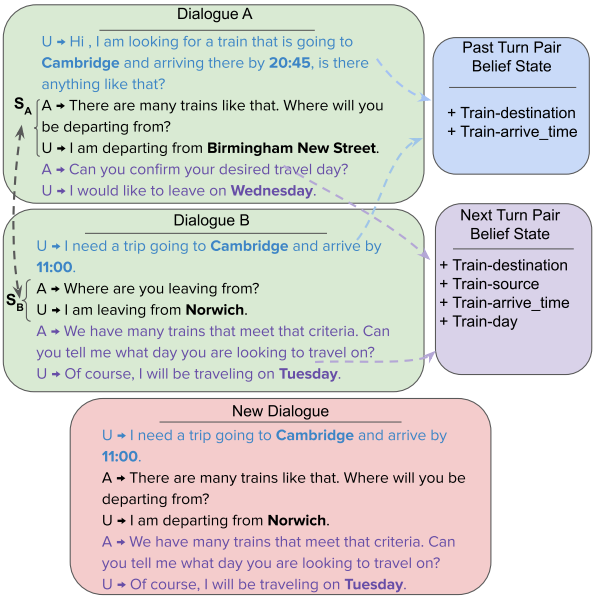

Figure 1: Scenario with two dialogues from *train* booking domain. Dialogue snippets, S_A&S_B, have the same dialogue function and the new dialogue created by replacing them and inserting proper slot values is still coherent end to end.

There have been several attempts to augment conversational data in the literature. Quan and Xiong (2019) up-sample the data through word or sentence level modifications, following standard text augmentation techniques in NLP such as synonym substitution, back-translation, or paraphrasing. Kurata et al. (2016) perturb embeddings of single utterances and decode similarly functioned synthetic utterances. Gao et al. (2020) create an end-to-end pipeline that finds the utterances with similar dialogue functions and trains a paraphrasing model. Most related to our method, CoCo (Li

et al., 2021) trains a conditional user–utterance generation model, then generates synthetic turns by modifying belief states using a rule-based system and conditioning the model on the modified belief state. Gritta et al. (2021) create a working graph of TOD datasets where each edge is a dialogue act and create synthetic dialogues by traversing alternative paths; however, their framework requires user acts to work with. Critically, none of the above techniques exploit the belief state annotations of TODs within an n-shot scenario.

In contrast, dialogue belief state annotations guide our approach to an effective n-shot augmentation method. We observe that the belief state identifies the specific slots that each turn-pair discusses. As such, belief states can be used as a proxy to represent dialogue function. Our method exploits this. For example, after exchanging two turn-pairs that serve the same dialogue function in two source dialogues, dialogue coherency should be preserved, if discounting necessary changes to slot values (Figure 1). Motivated by this, we delexicalize and store each turn-pair with their dialogue function to effectively construct new dialogues from scratch.

We evaluate our framework with MultiWOZ, a multi-domain dialogue dataset (Budzianowski et al., 2018). Each of its 10,000 dialogues is annotated with its turn belief states, system acts, and turn slots. We experiment using both the previous state-of-the-art (SOTA) recurrent TRADE (Wu et al., 2019) model and the transformer-based TOD-BERT (Wu et al., 2020b) model. Our framework significantly increases n-shot performance, both when adapting a DST model to a new domain and when adapting a language model to the DST task. A fine-grained analysis of evaluation results reveals that models finetuned on synthetic data become robust to previously unseen slot values, and recognize seen values better. The latter aspect accounts for the majority of the performance gain.

2 Related Work

2.1 Dialogue State Tracking

Previous DST models cumulatively keep track of utterances to obtain dialogue states (Williams and Young, 2007; Thomson and Young, 2010; Wang and Lemon, 2013). Lei et al. (2018) introduced Sequicity to generate belief spans as an intermediate process and improve the performance on the end task. Zhong et al. (2018) proposed to use a unique module for each slot, which improves the tracking of unseen slot values. The majority of these systems relied on an in-domain vocabulary and they were all evaluated on a single domain. Ramadan et al. (2018) proposed to jointly train the domain and state tracker using multiple bi-LSTMs and allowed the learned parameters to be shared across domains; whereas Rastogi et al. (2017) used a multi-domain approach using bi-GRU where the dialogue states are defined as distributions over a candidate set derived from dialogue history.

We use two base models in this paper. The first one, TRADE, was proposed by Wu et al. (2019). It implements an encoder–decoder architecture and applies a copy mechanism that helps to overcome out of vocabulary (OOV) challenges. The second one, TOD-BERT (Wu et al., 2020b), a task-oriented dialogue model following the transformer paradigm. It is pretrained using 9 TOD datasets with a contrastive objective function.

2.2 Few-shot Dialogue State Tracking

Many papers focus on the low-resource scenario in the DST field aiming to generate comparable results between low- and rich-resource settings. Existing studies invariably categorize into two approaches to address the low-resource challenge: (1) optimization functions aimed to exploit the smaller amounts of data, or (2) augmentation of the target data.

Few-shot Models and Techniques. Some approaches in the first class of solutions benefit from the recent transformer trend. One such study finetunes the GPT-2 model and reports n-shot slot-filling and intent recognition results on the SNIPS dataset (Madotto et al., 2020). They achieve promising results compared to baselines with fewer shots. TOD-BERT reports results on four downstream tasks in the full- and low-resource settings (Wu et al., 2020b). Another line of research tries to address the problem without transformers. Span-ConverRT re-defines the slot-filling problem as turn-based span extraction that helps greatly in the few-shot setting (Coope et al., 2020). Huang et al. (2020) use the model agnostic meta-learning (MAML) algorithm to adapt to new domains and show that it can outperform traditional methods with fewer data. Coach (Liu et al., 2020), on the other hand, breaks the slot-filling task into two components: a first slot entity detection task, followed by an entity type prediction task.

Data Augmentation for the Few-shot Setting. Other studies, like our approach, focus on augmentation to improve few-shot performance. Quan and Xiong (2019) adopt four techniques for augmentation: synonym substitution, stop-word deletion, translation and paraphrasing at sentence level. Kurata et al. (2016) start by pretraining a dialogue encoder–decoder, and then perturb the dialogue representations to back-decode synthetic dialogues. Another study by Jalalvand et al. (2018) trains a logistic regression model on the small target data to detect the most informative n-grams and then find related samples from an out-of-domain corpus. Yin et al. (2020) propose a reinforcement learning setting, alternating learning between a generator and a state tracker to discover augmentation policies that benefit the end task. Two separate studies try to solve the OOV problem by enriching dialogue slot values with other values (Song et al., 2020; Summerville et al., 2020).

Campagna et al. (2020) create an abstract dialogue model by defining domain templates through manual observations and then generate augmented data using these templates. Their model improves the zero-shot performance, but requires manual work for each new domain.

Three studies use dialogue annotations during the augmentation process. PARG matches turns of a task-oriented dialogue by their dialogue state to create pairs for paraphrase generation (Gao et al., 2020), they then jointly train the paraphrase generator with the end task outperforming other dialogue augmentation baselines. The low-resource setting defined by PARG is still required to be large enough to train a neural paraphrase model from scratch, thus limiting its applicability to emerging domains with little data. Moreover, they do not model the interaction of a turn-pair with the next turn-pairs; as such a paraphrased utterance may be noisy, repeating a slot on the next turn. Gritta et al. (2021) create graph representations of dialogue datasets where each edge corresponds to a dialogue act by the user or system. They then extract alternative dialogues. However, they experiment only using full data settings. Additionally, their framework presumes the dialogue states are specific to each utterance, but for MultiWOZ (among other datasets) dialogue states harbor information from a pair of system–user utterances. Lastly, Li et al. (2021) train a conditional user-utterance generation model on a large labeled dataset, then generate synthetic dialogues by mutating the belief states through a rule-based system. This method is also limited as it requires enough data to train a conditional generation model, an unrealistic requirement for few-shot training.

3 Method

Our method leverages a simple hypothesis, visualized in Figure 1: that the function of a pair of turns in a dialogue can be defined by its slots, and its interactions with its previous and next turn-pairs. The example has two turn-pairs: S_a from Dialogue A and S_b from Dialogue B. The turn-pair belief states that precede both S_a and S_b are composed of the same set of slot labels. The same holds for the belief states of turn-pairs following S_a and S_b. Their function in the dialogue is the same. We hypothesize the interchange of these pairs of turns (after changing the values according to the parent dialogue state) maintains a coherent dialogue. Our observations on the MultiWOZ dataset showed that this is true to a large extent for task-oriented dialogues because the belief state history represents the ongoing topic, and slot labels of the next turn give hints about the system acts.

Our framework implements this hypothesis in three steps. In Step 1 (§ 3.1), we create turn-pair templates by delexicalizing each pair (replacing slot values with their respective slot label), then storing each template with the previous, current, and next pair's belief states (*cf.* Figure 2). We also mine a dictionary of possible slot label–value pairs to be used in filling generated templates. In Step 2 (§ 3.2) we create dialogue templates by combining these pairs constrained such that two consecutive pairs' dialogue functions do not break coherency. We do this combination in a breadth-first manner, best visualized as a tree where each node is a turn-pair template, and every string of nodes from root to leaf is a dialogue template (*cf.* Figure 3). Finally in Step 3 (§ 3.3), we create final synthetic dialogues by filling the slot labels in the dialogue templates (*cf.* Figure 4) using the mined dictionary.

3.1 Step 1: Turn-pair Template Generation

Figure 2 depicts a sample turn-pair template that our framework generates. Each turn-pair template in our framework consists of a pair of turns: a system turn and a user turn. Our templates consist of pairs of turns, simply because consecutive turns (system–user) share the same dialogue state

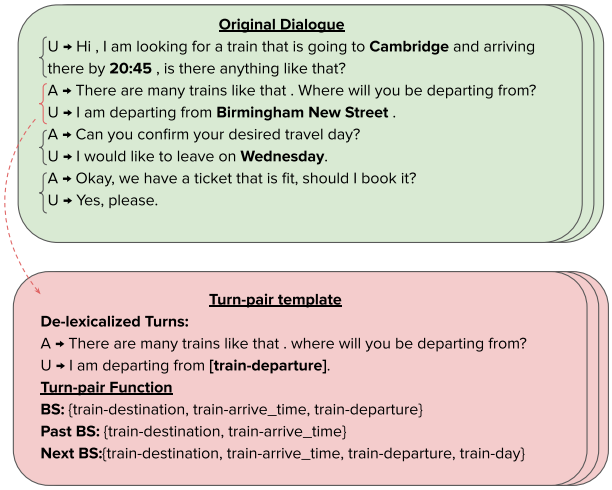

Figure 2: Sample turn-pair template (bottom, pink) and the original dialogue it is extracted from (top, green). The subject template is composed of four elements: 1) delexicalized turn utterances, and the belief state of 2) current, 3) past, and 4) next turns in the original dialogue.

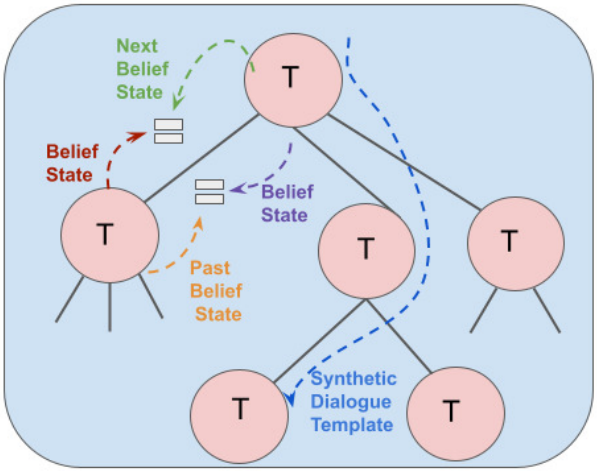

Figure 3: In our framework, dialogue templates are generated through adding proper turn-pair templates in a chain structure. The chains form a tree, which covers every possible dialogue template as a path from root to a leaf node.

annotation. Each turn-pair template consists of a delexicalized pair of turns and a dialogue function formed as the combination of the previous, current, and next turn belief states.

During delexicalization we follow (Hou et al., 2018) to replace each slot value with "[slot-name]". Since MultiWOZ 2.1 does not provide indices for slot values, we manually find each value by searching in the turn-pair. This brings up several problems where two slots might have the same value or where some categorical values might not show up in the text (*e.g. hotel-internet*: {"dontcare", "yes", "no"}). We filter out templates with the same values for different labels and leave the values for the categorical labels the same, assuming that they are independent of changes in other values. However, unlike non-categorical ones, we are limited from enriching the values of such slot types in our last step, surface realization, when we fill in our templates.

Each dialogue in MultiWOZ usually starts with a salutation and ends with a farewell. To distinguish these starting–ending pairs, we define two exception cases: (1) If a template's turn-pair comes from the beginning of a dialogue, we set its previous belief state as *null* (start state), (2) if it comes from the ending of a dialogue we set its next belief state as *null* (end state). We use these two cases later in template generation to generate coherent dialogues from start to end.

3.2 Step 2: Dialogue Template Generation

We generate each dialogue template by combining a set of turn-pair templates. We form our dialogue templates using a tree structure where each node corresponds to a turn-pair template, and a chain of nodes starting from a root and ending with a leaf is a dialogue template (Figure 3). We start by defining a root node and setting its belief state as *null*. Initially, we ignore the next belief state condition and add every template whose previous belief state is *null* — such turns are legitimate conversation starters (roots). At each level, we mark every newly-added node as an active node. Then after each level, we iterate through active nodes and expand each node with the set of eligible templates. Two conditions need to be met to append Template B to the tail of Template A: (1) B's belief state slots should be met by A's next belief state slots and (2) A's belief state slots should be met by B's past belief state slots. We continue adding templates until there are no active nodes. Eventually, we end up with a tree structure where each connected node represents a turn-pair and each path from the root to a leaf node is a unique dialogue template. We discard paths whose leaf nodes do not have *null* as the next belief state. This ensures that the dialogue template has a valid ending.

3.3 Step 3: Surface Realization

We now fill in the delexicalized dialogue templates. Using the slot–value dictionary extracted in Step 1, we fill each dialogue with every possible slot value combination thus effectively sourcing synthetic

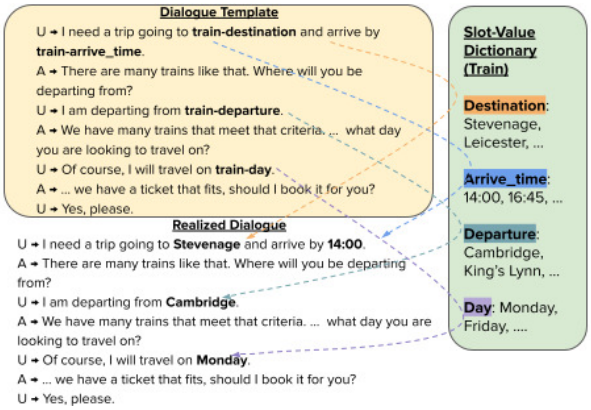

Figure 4: The last step in our framework, surface realization, utilizes the dictionary of slot label and slot values obtained from the original dialogues in Step 1, populating the templates with every permutation of possible values of each slot.

augmented dialogues (Figure 4). This final step returns a set of task-oriented dialogues, suitable for training (or fine-tuning) a learning system (*cf.* Appendix A for sample dialogues).

4 Experiments

4.1 Dataset, Models and Evaluation

We conduct experiments on MultiWOZ, a well-known dataset in the DST field. When compared to its counterparts like WOZ (Wen et al., 2017), DSTC2 (Henderson et al., 2014) and Restaurant-8k (Coope et al., 2020), MultiWOZ is the richest, combining several domains with a variety of slot labels and values. MultiWOZ is a multi-domain dialogue dataset that covers 10,000 dialogues between clerks and tourists, each annotated with turn belief states, system acts, and turn slots. Following prior works (Wu et al., 2019, 2020a) we conduct our experiments on 5 of 7 domains leaving *hospital* and *police* domains out as the sample quantity is very low in validation and test sets of these domains.

We wish to assess how fine-tuning with our augmented data affects model performance. We experiment with the TRADE and TOD-BERT models (Wu et al., 2020a, 2019) to assess whether their base performance can be improved using our augmentation framework. For both models, we follow the fine-tuning experiments done by (Wu et al., 2019): we train a base model on four domains and then fine-tune this model with small sets of randomly sampled data from the remaining left-out target domain (5- or 10-shots). We compare this against the scenario where we apply our augmentation framework on the small set before fine-tuning. Due to space limitations, we present results only for the subset of the *restaurant*, *taxi*, and *hotel* domains in TOD-BERT. These three domains cover almost every unique slot in the MultiWOZ dataset, and is thus representative. We conduct an additional experiment for TOD-BERT, training/testing with data from all domains in several few shot settings (20-, 40-, and 80-shot).

We evaluate TRADE using the metrics proposed by Wu et al. (2019): Slot Accuracy and Joint Accuracy. *Slot Accuracy* measures the proportion of correctly predicted slot values; while *Joint Accuracy* is more coarse-grained, measuring the correctly predicted turn dialogue states. To predict a turn dialogue state correctly means that all its contained slot values are predicted correctly. Also, when a slot is not mentioned in the utterance the ground truth for that slot becomes *None*. This results in utterances having ground truth slot values which mostly consist of the value *None*. We observe that in our few-shot experiments, unlike TRADE, TOD-BERT model returns predictions consisting only of *None* values. We believe that the discrepancy is attributable to TRADE's copy mechanism, which the TOD-BERT model lacks. To better assess the contribution of our augmentation approach, we use Active Slot Accuracy (Dingliwal et al., 2021) for the TOD-BERT experiments, which is the accuracy of slot value predictions for all non-*None* values.

4.2 Implementation and Training Settings

We adjust our training settings to facilitate a fair comparison among the models trained on different data sizes (original versus augmented). For the TRADE model, we use the default hyperparameter settings reported in the original paper. For TOD-BERT, we change the training batch size to 4 and the evaluation batch size to 8, the development set evaluation frequency to 1 evaluation per 200 steps, set the terminating condition to early stopping bounded by maximum number of steps. For our augmented fine-tuning model training, we fine-tune the base model on synthetic data for $N/2$ steps, followed by fine-tuning on the mixture of original and synthetic data for another $N/2$ steps. We perform this mixing of original samples in the latter part of fine-tuning to ensure that the model is exposed to a diverse set of samples, while not significantly deviating from the original distribution. This is conceptually similar to the notion of experience replay in reinforcement learning.

	Hotel		Taxi		Restaurant		Attraction		Train	
	Joint	Slot	Joint	Slot	Joint	Slot	Joint	Slot	Joint	Slot
1. Base Model (BM) trained on other 4 domains	0.12	0.64	0.60	0.73	0.12	0.54	0.18	0.54	0.22	0.49
2. BM fine tuned with 1% data (~80 dialogues)	0.21	0.76	0.61	0.75	0.21	0.77	0.43	0.74	0.61	0.91
5-Shot Augmentation on Target Domain										
3. BM fine-tuned with 5 samples	0.12	0.65	0.59	0.75	0.12	0.58	0.25	0.59	0.25	0.66
4. BM fine-tuned with augmented samples	0.12	**0.67***	0.58	0.75	**0.13**	**0.62***	**0.26**	**0.61**	**0.31***	**0.77***
10-Shot Augmentation on Target Domain										
5. BM fine-tuned with 10 samples	0.14	0.68	0.60	**0.76**	0.13	0.63	0.30	0.63	0.37	0.81
6. BM fine-tuned with augmented samples	**0.15**	**0.69**	0.60	**0.76**	**0.16***	**0.70***	**0.32***	**0.66***	**0.39**	**0.83**

Table 1: Evaluation results of TRADE model. The first row shows the zero shot results; the second row, the finetuning with 1% data (80 dialogues) for comparison with n-shot results. Each figure is an average of 10 runs. **Bolded** numbers in each section shows the best performance within that section. "*" indicates statistically significant results with 95% confidence.

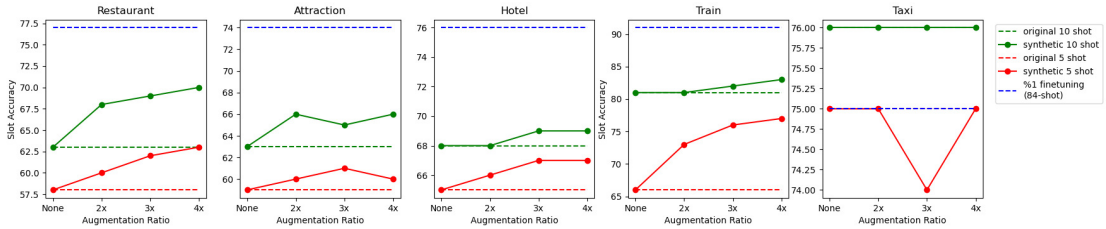

Figure 5: Effects of the augmentation ratio on TRADE model by domain. The dashed blue line represents the performance of fine-tuning with 1% of full data (~80 dialogues) for comparison as a pseudo upper bound [Note y-axis scales differ per chart].

Active Slot F1	Restaurant	Taxi	Hotel
5-Shot			
3'. Original	0.16	0.0065	0.20
4'. Augmented	**0.19***	**0.0078**	**0.22***
10-Shot			
5'. Original	0.20	0.010	0.18
6'. Augmented	**0.22***	**0.013***	**0.23***

Table 2: TOD-BERT evaluation results over the individual *restaurant*, *taxi* and *hotel* domains, averaged over 10 runs.Best performance within each shot level are **bolded**; statistical significance ($p \geq 95\%$) is starred.

Active Slot F1	20-shot	40-shot	80-shot
Original samples	0.10	0.16	0.21
Our augmented samples	**0.16***	**0.21***	**0.24***

Table 3: TOD-BERT evaluation results over all domains, averaging 10 runs. Best performance within each shot level are **bolded**; statistical significance ($p \geq 95\%$) is starred.

4.3 Results

TRADE Experiments (Table 1). We report the significance of results with 95% confidence along with averages over 10 runs. Our framework can sustain the model performance in all five domains, and significantly improves over baseline (Row 1) in either the 5- (Row 4) or 10-shot (Row 6) scenarios in four of the five domains, where most results are statistically significant at the $p \geq 0.95$ level. These results also greatly improve over fine-tuning using just 5 or 10 target domain samples (compare Row 3 against 4, and Row 5 against 6). Overall, applying our augmentation framework yields a macro-averaged improvement of 3.2% slot accuracy and 1.5% joint accuracy. As a pseudo-upper bound, we compare our method against fine-tuning over 80 shots (roughly 1% of the target domain data, represented by Row 2), and see that our approach significantly closes this performance gap.

The exception is the *taxi* domain where the augmented data does not result in significant change. We believe this is due to *taxi* domain slots having higher variety in values than in other domain slots. This results in many OOV values in the test set. The TRADE model, thanks to its copy mechanism, adapts well to these OOV with fewer data. The fact that the performance of the base model fine-tuned with 1% of data is already reached by fine-tuning the same model within a 5-shot scenario (compare Row 2 and Row 1's *taxi* column) supports our claim.

TOD-BERT Experiments (Tables 2 and 3). With TOD-BERT, we examine our framework's effect on both domain and task adaptation.

· Table 2 shows results for domain adaptation, and the figures are comparable to the those in Table 1 for TRADE. We number the rows with primes ($'$) to imply the comparable results from the TRADE experiments. We follow the same setting as above for TRADE (train on 4 other domains, test on target domain). We observe uniformly improved results

Recall	Unseen Values	Seen Values
All-domains		
Original	0.1 e-3	0.24
Augmented	**0.2 e-3**	**0.28**
Restaurant		
Original	1.5 e-3	0.20
Augmented	**2.3 e-3**	**0.26**
Taxi		
Original	6.3 e-3	0.16
Augmented	**6.8 e-3**	**0.21**
Hotel		
Original	0.5 e-3	0.30
Augmented	**1.0 e-3**	**0.32**

Table 4: TOD-BERT evaluation results, subdivided between on seen and unseen values, averaged over 10 runs, with best results per section in **bold.**

Error type	Original	Synthetic
restaurant-food	2,041	1,675
restaurant-pricerange	1,210	603
restaurant-name	1,133	1,061
restaurant-area	853	480
restaurant-book day	743	335
restaurant-book people	740	212
restaurant-book time	1,119	347

Table 5: Fine-grained *restaurant* domain errors, for the original and augmented TRADE model, classified by slot type.

over the few shot fine-tuned models, as we did for TRADE, proving agnostic feature of our framework.

· Table 3 shows results for task adaptation. Here, the TOD-BERT model has no familiarity with the DST task at all, thus fine-tuning is an adaptation to the task itself. This is a more challenging scenario. Again, we see uniform improvement, especially marked for the lower-shot scenarios (20- and 40-).

The results for both are consist and in favor of our framework. Our framework helps in both cases: (1) LM adaptation to a new task (*e.g.* DST), and (2) LM adaptation to a new task-oriented dialogue domain (*e.g. restaurant*).

4.4 How Does Augmentation Improve Performance?

To study the reason behind the performance gain by augmentation, we dispart our test set samples into two groups: samples with unique values that do not show up during training, and samples with values seen during training. We then evaluate the TOD-BERT model trained with original and synthetic data on these two separate groups, *cf.* Table 4. The results suggest that although, augmentation increases robustness to unseen values in all domains, the largest part of the contribution is on seen values. This is expected since our framework uses the same set of values as in small original dialogue set during surface realization

Note that for the "All-domains" section in the table the improvement on unseen values is smaller compared to domain-specific sections (*Restaurant*, *Taxi*, *Hotel*), this is because, in the former, the model learns DST task from scratch thus exploiting seen-values to learn the task overweighs to generalizing over unseen values. Whereas for the latter, robustness to unseen values gets higher learning priority since the model is already familiar with the DST task from training on other 4 domains.

This analysis shows that our framework helps the model to exploit slots that have a bounded value pool with less unique values while also making it robust to unseen values for slots with broader value pools.

4.5 Effect of Augmentation Ratio

We run our framework with several different augmentation ratios in both the 5 and 10 shot cases to inspect if the synthetic data amount affects the results proportionally. Figure 5 shows the results for the TRADE model in all 5 domains. Our framework increases the results steadily compared to base fine-tuning, and the amount of synthetic data affects the results proportionally in almost every case except the *taxi* domain as explained before (*cf.* Section 4.3).

4.6 Fine-grained Error Analysis

4.6.1 Slot-type Errors

Apart from performance in evaluation metrics we also analyze the error rates of the TRADE model in each specific slot type in the *restaurant* domain and compare results with and without our framework. Table 5 shows the results. Our framework consistently reduces error rates in every single slot type. The drop in the error rate is least remarkable for the name and food slots, we believe this is because the challenge in these slots is most largely unknown vocabulary words. Our framework enriches the dialogue templates with values from the original set. Thus it is less helpful for those slots suffering from the unknown slot value problem and shows more significant improvements on slots with arguably more isolated vocabulary (*e.g.* Book-day: 1, 2, 3, *etc.* or price range: cheap, moderate, expensive).

To support the significance of results on fine-grained slot error types, we use McNemar's test

	Hotel		Taxi		Restaurant		Attraction		Train	
	Joint	Slot	Joint	Slot	Joint	Slot	Joint	Slot	Joint	Slot
5 Shot Augmentation on Target Domain										
BM fine-tuned with CoCo	0.12	0.66	**0.60**	0.75	**0.13**	**0.62**	0.24	0.58	0.27	0.69
BM fine-tuned with our framework	0.12	**0.67**	0.58	0.75	**0.13**	**0.62**	**0.26**	**0.61**	**0.31**	**0.77**
10 Shot Augmentation on Target Domain										
BM fine-tuned with CoCo	**0.15**	0.68	**0.61**	0.75	**0.16**	0.67	0.31	0.64	**0.39**	0.82
BM fine-tuned with our framework	**0.15**	**0.69**	0.60	**0.76**	**0.16**	**0.70**	**0.32**	**0.66**	**0.39**	**0.83**

Table 6: Evaluation results of TRADE model comparing our augmentation framework to the upperbound CoCo model pre-trained on full training data (including target domain).

Active Slot F1	Restaurant	Taxi	Hotel
5 Shot			
CoCo	0.17	0.0047	0.21
Ours	**0.19**	**0.0078**	**0.22**
10 Shot			
CoCo	**0.22**	0.0114	0.21
Ours	**0.22**	**0.0132**	**0.23**

Table 7: Evaluation results of TOD-BERT model comparing our augmentation framework to the upperbound CoCo model pre-trained on full training data (including target domain).

Active Slot F1	Restaurant	Hotel
5 Shot		
Full pipeline	**0.183**	**0.255**
Only SR	0.157	0.250
10 Shot		
Full pipeline	0.198	**0.258**
Only SR	**0.237**	0.243

Table 8: TOD-BERT target domain experiments comparing full pipeline (first row) against only surface realization (second row). Each number corresponds to an average of 3 runs.

($\alpha = 0.01$) upon creating the confusion matrix between our framework and original fine-tuning. The results suggest that synthetic data fine-tuning shows statistically significant improvements over the original data fine-tuning, with $p < \alpha$.

4.7 Comparison against CoCo Model

To better locate the position of our framework in the literature we repeat target domain experiments using another dialogue augmentation method: CoCo. However, CoCo is a learning-based approach that requires rich amounts of data, so it is unfair to expect it to learn from only a few shots (5/10). Instead, we use the pretrained weights that are provided by the original CoCo paper and treat it as an upper bound because it is trained on the full training data (including the target domain for leave-one-out experiments) whereas our framework uses only the provided few dialogues during augmentation.

Tables 6 and 7 give the results for TRADE and TOD-BERT, respectively. Despite the advantageous standing of CoCo, our framework outperforms CoCo in all domains for the TOD-BERT model and shows either superior or comparable results on TRADE.

4.8 Effect of Template Generation

We conduct an ablation study to show the contribution of our novel dialogue template generation idea by re-running the TOD-BERT target domain experiments for *hotel* and *restaurant* domains with a simpler baseline, where we use only the original n dialogues as templates and perform surface realization.

The results in Table 8 show that template generation improves results compared to only performing surface realization in most of the cases. Our template generation strategy offers higher diversity to the samples but it might bring up noisy samples along, whereas only surface realization is less noisy but lacks the diversity that novel templates contribute.

5 Conclusion

We introduced a task-oriented dialogue augmentation framework that exploits embedded information in dialogue states. We experimented on the MultiWOZ dataset using both the TRADE model as representative of sequential models and TOD-BERT, a transformer-based BERT model adapted for task-oriented dialogues. Evaluation results suggest that our framework improve the performance when adapting a DST model to a new, unseen domain and also when adapting an LM to the DST task. Empirical analysis shows that although our augmentation increases performance on unseen values, a large portion of its boost comes from performance on seen values. We conclude that our framework consistently improves the few-shot performance over the DST task and can open doors for many other TOD tasks in limited data scenarios.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

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*, pages 122–132, Online. Association for Computational Linguistics.

Samuel Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson. 2020. Span-ConveRT: Few-shot span extraction for dialog with pretrained conversational representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 107–121, Online. Association for Computational Linguistics.

Saket Dingliwal, Shuyang Gao, Sanchit Agarwal, Chien-Wei Lin, Tagyoung Chung, and Dilek Hakkani-Tur. 2021. Few shot dialogue state tracking using meta-learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1730–1739, Online. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 639–649, Online. Association for Computational Linguistics.

Milan Gritta, Gerasimos Lampouras, and Ignacio Iacobacci. 2021. Conversation graph: Data augmentation, training, and evaluation for non-deterministic dialogue management. *Transactions of the Association for Computational Linguistics*, 9:36–52.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2014. The Second Dialog State Tracking Challenge. Technical report.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1234–1245, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yi Huang, Junlan Feng, Min Hu, Xiaoting Wu, Xiaoyu Du, and Shuo Ma. 2020. Meta-Reinforced Multi-Domain State Generator for Dialogue Systems. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7109–7118.

Shahab Jalalvand, Andrej Ljolje, and Srinivas Bangalore. 2018. Automatic data expansion for customer-care spoken language understanding. *CoRR*, abs/1810.00670.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Gakuto Kurata, Bing Xiang, and Bowen Zhou. 2016. Labeled Data Generation with Encoder-decoder LSTM for Semantic Slot Filling.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1437–1447, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2021. Coco: Controllable counterfactuals for evaluating dialogue state trackers. In *International Conference on Learning Representations*.

P. Liu, X. Wang, C. Xiang, and W. Meng. 2020. A survey of text data augmentation. In *2020 International Conference on Computer Communication and Network Security (CCNS)*, pages 191–195.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A coarse-to-fine approach for cross-domain slot filling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 19–25, Online. Association for Computational Linguistics.

Andrea Madotto, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Language models as few-shot learner for task-oriented dialogue systems.

Jun Quan and Deyi Xiong. 2019. Effective data augmentation approaches to end-to-end task-oriented dialogue. *2019 International Conference on Asian Language Processing (IALP)*, pages 47–52.

Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-Scale Multi-Domain Belief Tracking with Knowledge Sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 432–437, Stroudsburg, PA, USA. Association for Computational Linguistics.

Abhinav Rastogi, Dilek Hakkani-Tur, and Larry Heck. 2017. Scalable Multi-Domain Dialogue State Tracking. *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017 - Proceedings*, 2018-January:561–568.

Xiaohui Song, Liangjun Zang, Yipeng Su, Xing Wu, Jizhong Han, and Songlin Hu. 2020. Data augmentation for copy-mechanism in dialogue state tracking. *CoRR*, abs/2002.09634.

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

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

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

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

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

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020a. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020b. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.

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

Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. 2017. Data noising as smoothing in neural network language models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dialog state tracking with reinforced data augmentation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9474–9481. AAAI Press.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *CoRR*, abs/1804.09541.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-Locally Self-Attentive Encoder for Dialogue State Tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, Stroudsburg, PA, USA. Association for Computational Linguistics.

A Sample Synthetic Dialogues

We showcase two synthetic dialogues generated with our framework, *cf.* Table 9, by merging templates from 2 and 3 different dialogues respectively. Our framework can generate multi-domain dialogues following the original dialogues' belief state. For instance, the first half of the left-hand side dialogue includes slots from the *restaurant* domain, while the second half has slots from the *train* domain. The right-hand side dialogue on the other hand combines slots from domains: *train* (from two different dialogues) and *hotel* (from another third dialogue). Although both dialogues seem coherent in shape, the latter has a redundancy where the system request the day information after the user already stated it. This is because of a missing annotation where the train-day slot in the belief state of the third turn is missing. These kinds of annotations unavoidable but negligible because it recaptures a misunderstanding by the agent which is observed in real dialogues frequently.

Speaker	Turn
User	I am coming to town and looking for a good Spanish restaurant to try near the town center. Do you have any suggestions ?
Agent	The Spanish restaurant near center is Wagamama. The address is 36 saint Andrews street .
User	I'll be needing a table for 4 at 5p.m. this next Monday , can you help ?
Agent	Here is your reference number 6huyuygs is there anything else I can do for you?
User	Yes I need to find a train to Cambridge from Kings Lynn.
Agent	When would you like to travel ?
User	Tuesday morning at 9 am
Agent	train tr3447 leaves Kings Lynn at 09:11. Would you like to reserve a ticket ?
User	Yes please. Make a booking for the same number of people .
Agent	Okay, the booking was successful , the total fee is 58.8 gbp payable at the station . reference number is : fz6dqhqt. Anything else i can help you with?
User	That is all i needed today.

Speaker	Turn
User	I am looking for a train to Cambridge that departs after 10:45.
Agent	What day would you like to travel?
User	I'll be travelling on a Wednesday.
Agent	Where would you be departing from?
User	I will be leaving Norwich.
Agent	On what day will you be leaving?
User	I will be leaving on Wednesday.
Agent	It is 18.88 pounds per ticket. How many tickets would you like to purchase?
User	Just 2 tickets please and thank you.
Agent	Your ticket has been reserved . Total fee is 18.88 gbp, payable at the station. Reference number zm5k0a4q. Can i help you with anything else today?
User	Can you book Acorn guest house for 7 people starting on Wednesday for 4 nights, please ?
Agent	Thank you so much for reserving my train ticket. I have the information for Acorn guest house you requested. Would you like me to reserve it for you ?
User	Yes. I need the reference number.
Agent	The booking was successful. your reference number is dtkw11yu
User	That is all I needed today, thank you for your help.

Table 9: Sample synthetic dialogues generated by our framework. (L) Dialogue in *restaurant* domain generated by 2 different original dialogues. (R) Dialogue in *train* domain generated by 3 different original dialogues.