A Simple But Effective Approach to n-shot Task-Oriented Dialogue Augmentation

Taha Aksu †‡, Nancy F. Chen †‡, Min-Yen Kan †, Liu Zhengyuan ‡
† National University of Singapore, Singapore
‡ Institute for Infocomm Research, A*STAR, Singapore
taksu@u.nus.edu,
nfychen@i2r.a-star.edu.sg,
kanmy@comp.nus.edu.sg,
liu_zhengyuan@i2r.a-star.edu.sg

Abstract

The collection and annotation of task-oriented conversational data is a costly and time-consuming manner. Many augmentation techniques have been proposed to improve the performance of state-of-the-art (SOTA) systems in new domains that lack the necessary amount of data for training. However, these augmentation techniques (e.g., paraphrasing) also require some mediocre amount of data since they use learning-based approaches. This makes using SOTA systems in emerging low-resource domains infeasible. We, to tackle this problem, introduce a framework, that creates synthetic task-oriented dialogues in a fully automatic manner, which operates with input sizes of as small as a few dialogues. Our framework uses the simple idea that each turn-pair in a task-oriented dialogue has a certain function and exploits this idea to mix them creating new dialogues. We evaluate our framework within a low-resource setting by integrating it with a SOTA model TRADE in the dialogue state tracking task and observe significant improvements in the fine-tuning scenarios in several domains. We conclude that this end-to-end dialogue augmentation framework can be a crucial tool for natural language understanding performance in emerging task-oriented dialogue domains.

1 Introduction

Data augmentation is a method used for increasing the diversity and size of a dataset by synthesizing new samples through applying transformations to the original data. This method has proven to be beneficial in image processing which includes 2D transformations on the image such as shifts, rotations, flips, etc. Text data, on the other hand, does not have a continuous space that can be manipulated with such transformations and unlike for an image, an atomic change in a sentence (e.g., a letter or a word) can result in a noisy sample. Thus, it requires much elaborative work to generate rational natural language phrases through data augmentation.

The majority of existent methods used for text augmentation are targeted towards written form of text (e.g., passages, news articles, etc.). These methods operate with word or sentence level mutations in the original text data creating new synthesized sentences/paragraphs. Another form of text modality popular in NLP comes in a conversational structure where two or more speakers exchange utterances. This study focuses on the augmentation of a specific type of conversational data: Task-oriented dialogues.

A task-oriented dialogue is a conversation between a user and an agent, where the user tries to achieve some end goal by listing his/her pref-
ferences, and the agent helps them by enquiring the user about their preferences and extracting related information. One of the most fundamental tasks within task-oriented dialogues, dialogue state tracking (DST), is to be able to detect these preferences in the dialogue. For this task each pair of utterances in a task-oriented dialogue is annotated with slot-label & slot-value pairs (e.g. train-destination: Cambridge, train-arrive-time: 20.45) and a belief state. A belief state is a dictionary that shows the final values of each slot label after the subject utterance.

There have been several attempts to augment conversational data in the past. Quan and Xiong [2019] up-sample the data by doing word or sentence level modifications. These modifications follow the standard text augmentation techniques in NLP such as synonym substitution, back-translation, or paraphrasing. Kurata et al. [2016] perturb and regenerate the representations of single utterances, effectively creating new utterances with similar functions. Gao et al. [2020] create an end-to-end pipeline which finds the utterances in a dialogue dataset that have similar dialogue functions and train a paraphrase model. They augment the dataset with the paraphrase model and train the end task jointly with the paraphrasing. They, however, fail to model the interaction of a turn-pair with the previous and next turn-pairs. Moreover, since they train a neural paraphrasing system from scratch they still require a mediocre amount of data in the target domain. None of the augmentation techniques introduced above exploit the belief state annotations of task-oriented dialogues which embeds the underlying linked structure of turn-pairs. These dialogue state annotations guide our approach to an effective task-oriented dialogue augmentation method. We show, in this study, that using turn-pair relations in a dialogue for augmentation leads to an efficient framework that works well with very low-scale data.

Our method is built upon a simple idea that makes use of the belief states of turn-pairs in a task-oriented dialogue. The belief state, simply put, is like a pointer to topics that each turn-pair discusses. Thus, we claim the dialogue states before and after a turn-pair can be used as an identifier to represent the function of that pair in the dialogue. And further claim that the coherency of a dialogue should not get harmed by replacing one of its turn pairs with another, given that they share the same dialogue function and necessary changes to slot values are made to fit the dialogue, cf. Figure 1. Motivated from this hypothesis we then delexicalize and store each turn-pair with those two pointers, and effectively reconstruct new dialogues from scratch.

We evaluate our framework with the well-known MultiWOZ dataset [Budzianowski et al., 2018]. MultiWOZ is a multi-domain dialogue dataset spanned over 7 different tasks. Each one of the 10,000 dialogues is annotated with turn-belief states, system acts, and turn slots. As we conduct our experiments with TRADE, a SOTA architecture from MultiWOZ leaderboard, following their paper we use only 5 of the 7 total domains because the other two domains are limited in the amount of data provided. Experiments show that our framework can significantly increase the performance in almost all domains. Specifically, we get 0.11, 0.7, 0.3, 0.2 improvements in slot accuracies (statistically significant) of train, restaurant, attraction, and hotel domains respectively within a few-shot setting. We observe that the only domain our framework can not improve, the taxi domain, does not favor even more original data because it performs the same given 84 dialogues (%1 of full data) and 5 dialogues from original data. This exceptional behavior naturally deemphasizes the use case of any augmentation method along with ours.

This work shows a different perspective for text augmentation in the task-oriented dialogue domain which differently from past studies exploits the conversational nature and interrelations in a dialogue. We believe that it can be a great tool if facilitated in limited data scenarios for task-oriented dialogue tasks and ease training data-hungry models in emerging domains.

2 Related Work

2.1 Dialogue State Tracking

Dialogue state tracking (DST) is one of the core tasks of today’s conversational systems where the challenge is to estimate the user’s preferences at every utterance of the dialogue. Past models in the field used to 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 end task. Zhong et al. [2018] proposed to use spe-
cial modules for each slot, which improve the tracking of unseen slot values. But the majority of these systems relied on an in-domain vocabulary and they are evaluated on a single domain. Ramadan et al. [2018] proposed to jointly train the domain tracker and state tracker using multiple bi-LSTM 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). The base model used in this paper is proposed by Wu et al. [2019], in their architecture they use a copy mechanism to overcome out of vocabulary (OOV) words, and they generate probability distributions over three cases for each slot: none, don’t-care, ptr.

2.2 Few-shot Dialogue State Tracking
Although there are many papers focused on low resource scenarios in the DST field, the scale of data that these papers define as “low” varies. Despite this variation, all studies are aimed towards comparable results between low and rich resource settings. Existing studies show two promising ways for addressing the low resource setting problem: (1) specialized models or adaptation techniques or (2) augmentation of target data.

2.2.1 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 report n-shot slot-filling and intent recognition results on SNIPS dataset [Madotto et al., 2020]. They achieve promising results compared to baselines with fewer shots, however, they also report that the max-input length of GPT-2 is a limitation over the number of max shots. TOD-BERT trains a language model on 9 task-oriented dialogue datasets and reports results on four downstream tasks in full and low resource settings [Wu et al., 2020]. Although powerful these systems prove to be costly to deploy and use. There is another line of research that tries to address the problem without using transformers. Span-ConverRT approaches the slot-filling problem as a turn-based span extraction problem which they claim helps greatly in the few-shot setting [Coope et al., 2020]. Huang et al. [2020] use the model agnostic meta-learning (MAML) algorithm for adapting to new domains and show that it is able to outperform traditional methods with less data. Coach on the other hand separates the slot-filling task into two by first detecting the slot entities in a sentence followed by the prediction of its entity type [Liu et al., 2020]. Their model is able to achieve better adaptation than existing baselines.

2.2.2 Data Augmentation For Few-shot Setting
The second class of research, and this paper, is focused on enriching the data in the target domain to improve the existing learning systems’ 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 pre-training a dialogue encoder-decoder, they then perturb the dialogue representations in their dataset and decode them back obtaining synthetic dialogues. Another study by Jalalvand et al. [2018] train a simple logistic regression model on the small target data detecting most informative n-grams and then finding samples from an out-of-domain corpus using these n-grams. One common thing that all these studies share is that they do not exploit the structure of dialogues and can be applied to any other text form. PARG on the other hand matches turns of a task-oriented dialogue by their dialogue state, effectively creating pairs for paraphrase generation [Gao et al., 2020], they then jointly train the paraphrase generator with the end task outperforming other dialogue augmentation baselines. However, the low-resource setting defined by PARS is still required to be large enough to train a paraphrase model from scratch, thus limiting its applicability to emerging domains.

3 Method
Our method rises upon a simple hypothesis, which Figure 1 visualizes through an example: We define the function of a pair of turns in a dialogue by its slots and interactions with previous and next pairs of turns. For example, in the figure, there are two turn-pairs $S_a$ from dialogue A and $S_b$ from dialogue B. Both of these share the same previous past belief state and same set of slots in the incoming turn. Thus their function in the dialogue is the same. We claim that one can replace these pairs of turns after changing the values according to the parent dialogue state and still assume an end-to-end coherent dialogue, like the one in the figure. Our observations on the MultiWOZ dataset
Figure 2: Sample Turn-pair template (bottom) and the original dialogue (top) it is extracted from. The subject template is composed of 5 elements: Delexicalized agent and user utterances, belief state of current and past turns in the original dialogue, and slot label & values of current and next turns in the original dialogue.

Figure 3: Dialogue templates in our framework are generated through adding proper turn-pair templates end-to-end in a tree structure. This tree eventually covers every probable dialogue template as a path from root to a leaf node.

Figure 4: The last step in our framework, surface realization, first creates a dictionary of slot label & slot values from original dialogues. It then populates the templates with every permutation of possible values of each slot thus effectively creating final synthetic dialogues.

3.1 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) in the MultiWOZ dataset share the same dialogue state annotation. Each turn-pair template consists of a delexicalized pair of turns, the belief state after these turns, the belief state before these turns, and the slot labels of these turns, and
Lastly the slot labels of the next turn from the original dialogue.

Our delexicalization, following prior work [Hou et al., 2018], consists of finding the slot values in the turn-pair and replacing them with “[slot-name]”. MultiWOZ dataset does not provide the indexes for the values thus 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 some categorical values might not show up in the text (e.g. hotel-internet: {don’tcare, yes, no}). We filter out templates with the same values for different labels and we leave the values for the categorical labels the same assuming the values of these categorical slots are not affected by changes in other values. However, unlike not-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 the MultiWOZ dataset usually starts with a salutation and ends with a farewell. To distinguish these pairs, we define two special cases: (1) If a template’s turn-pair comes from the beginning of a dialogue then we set its previous belief state to be “null”, (2) if it comes from the ending of a dialogue then we set its next slots to be “null”. We use these two cases later in template generation to generate dialogues that are coherent from start to end.

### 3.2 Dialogue Template Generation

During the generation, we form each dialogue template by combining a set of turn-pair templates end-to-end. The combination is done with respect to the previous belief state and next slot labels of each template. These are respectively backward and forward limitations for adding a turn pair template to an on-going dialogue template. We form our dialogue templates using a tree structure where each node corresponds to a turn pair template and a string of nodes from root to leaf is a dialogue template, with the condition that the subject leaf node’s next slot label is null. Our generation process is shown in Figure 3. We start by defining a root node and setting its belief state as null. For the initial add, we ignore the next slot limitation thus adding every template whose previous belief state is null. 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 templates that they can be followed with. Two conditions need to be met to add template A to the tail of template B: (1) A’s next slot labels should be met by B’s labels and B’s past belief state labels should be met by A’s belief state. 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. Each path from the root to a leaf node in this tree is a unique dialogue template. We filter these with only the ones whose leaf nodes have null as the next slots. This ensures that the dialogue template has a legitimate ending such as a farewell.

### 3.3 Surface Realization

The last step in our method is to fill the dialogue templates. This step uses the slot value dictionaries for each slot label, which is extracted back in the initial step. Using this dictionary, we fill each dialogue with every possible combination of slot values thus effectively sourcing synthetic augmented dialogues in an end-to-end manner, cf. Figure 4. This final step brings us to the end of our method, returning a set of task-oriented dialogues fully trainable with a learning system.

### 4 Experiments

#### 4.1 Dataset 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 variety of slot labels and values. MultiWOZ is a multi-domain dialogue dataset that covers 10,000 dialogues between clerks and tourists at an information center over 7 distinct domains such as restaurant, taxi, ho-

| #dialogues | Hotel | Taxi | Rest | All | Train |
|------------|-------|------|------|-----|-------|
| 2191       | 2056  | 4688 | 3513 | 4081|
| #turn / #dial | 12.3 | 10.8 | 11.1 | 10.6 | 11.4 |
| #value / #slot | 16.7 | 194.7 | 55.14 | 56.3 | 56.8 |
| Slots       | price, | price, | price, | area, | name, |
|             | type, | day, | dest., | area, | depart., |
|             | area, | name, | arrive, | day, | time, |
|             | stay, | name, | arrive, | name, | people |
|             | stars, | day, | arrive, | people, | people |
|             | internet, | day, | arrive, | depart., | leave |
|             | park, | day, | arrive, | depart., | name |
|             | people | day, | people, | leave | people |

Table 1: MultiWOZ dataset statistics categorized by domain.
Table 2: Evaluation results for 5 and 10 shot finetuning with TRADE model with and without augmented data. First row shows the zero shot results, second row the finetuning with %1 data (80 dialogues) for comparison with n-shot results. Each number in the 5 and 10 shot sections is an average of 10 runs. For each run we randomly sample different dialogues from the dataset and augment them using our framework. Bolded numbers in each section shows the best performance within that section and the * indicates statistically significant results with %95 confidence.

| Domain       | Hotel Joint | Taxi Joint | Restaurant Joint | Attraction Joint | Train Joint |
|--------------|-------------|------------|------------------|------------------|-------------|
| Base Model 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 |
| 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 | 0.12 0.65 | 0.59 0.75 | 0.12 0.58 | 0.25 0.59 | 0.25 0.66 |
| BM fine-tuned with 5 samples | 0.12 0.67* | 0.58 0.75 | 0.13 0.62* | 0.26 0.61 | 0.31* 0.77* |
| 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 | 0.14 0.68 | 0.6 0.76 | 0.13 0.63 | 0.30 0.63 | 0.37 0.81 |
| BM fine-tuned with 10 samples | 0.15 0.69 | 0.6 0.76 | 0.16* 0.70* | 0.32* 0.66* | 0.39 0.83 |
| BM fine-tuned with augmented samples | 0.15 0.69 | 0.6 0.76 | 0.16* 0.70* | 0.32* 0.66* | 0.39 0.83 |

tel, etc. Each dialogue is annotated with turn belief states, system acts, and turn slots. The numbers for training, development, and testing data samples are respectively 8438, 1000, 1000. Table 1 shows further details over each domain in the MultiWOZ dataset. We evaluate the model following the metrics used within the TRADE model [Wu et al., 2019]: Slot Accuracy and Joint Accuracy. Slot accuracy stands for the proportion of correctly predicted slot values whereas joint accuracy for the correctly predicted turn dialogue states. For a turn dialogue state to be predicted correctly all of the slot values it includes have to be predicted correctly.

### 4.2 Implementation Details

We use the TRADE [Wu et al., 2019] model as a base model to finetune with our augmented data, which is one of the several SOTA models in the MultiWOZ leader-board. TRADE model is motivated in solving the unseen slot value problem by establishing a copy-mechanism that generates dialogue states from utterances while also exploiting the interrelations of domains through knowledge-transfer. Following TRADE we conduct our experiments on 5 of 7 domains (hotel, taxi, restaurant, attraction, and train).

Because we are interested in the transfer quality between domains using augmented data, we follow the fine-tuning experiments done in the TRADE paper. Specifically for each new domain, we train a base model on 4 other domains. We then sample a small set in the target domain and fine-tune the base model with the sampled dialogues. And finally, we augment the small sample of dialogues using our framework and fine-tune the base model on this synthetically augmented dataset. We then, compare the results of original and augmented fine-tuning.

### 4.3 Results

Table 2 shows the results of our experiments. The results for the base model are from a single run, whereas reported results for finetuned models are an average of 10 runs. Specifically for each n-shot scenario, we sample 10 n-dialogue sets from the original dataset and create 10 separate augmented sets using their corresponding original dialogues. We also report the significance of results with %95 confidence along with averages in the table. Our framework can sustain the performance of the model in every domain and for 4 out 5 of them the it significantly improves the results in either 5 or 10 shot scenarios. The only exception is the taxi domain where the additional augmented data does not do any significant change in both scenarios. We believe the reason behind this is that the dialogues in the taxi domain are overall pretty simple and follow a common pattern. The fact that the performance of the base model finetuned with %1 of data is already reached by finetuning the same model with 5 random dialogues (cf. Table 4.3) supports our claim. Like any augmentation framework, our system cannot generate better quality dialogues than the ones existent in the original distribution, thus it seems unrealistic to get performance improvements in an n-shot scenario on the taxi domain, using the TRADE model.

### 4.4 Effect of Augmentation Ratio

We have tried our augmentation scheme with several different augmentation ratios in the 5 and 10 shot cases to inspect if the synthetic data amount affects the results proportionally. Figure 5 shows
the results 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). For the few cases where the performance drops with increasing synthetic data, we verified through statistical testing that the differences are not significant with %95 confidence.

Still, we think that these slight drops in performance might be happening due to overfitting to certain template types. Because the number of templates that can be generated from only 5/10 dialogues is limited, the exposure to the same type of templates increases as the augmentation ratio goes up. Thus making the system overfit to those. We believe this could be addressed through filtering the synthetic dialogues by prioritizing those that the end system shows less confidence in. We hope to investigate this further in future work.

4.5 Fine-grained Error Analysis
4.5.1 Slot-type Errors
Apart from performance in evaluation metrics we also analyze the error rates of our framework in each specific slot type in restaurant domain and compare it to the model fine-tuned with original data. Table 3 shows the results. Our framework consistently reduces error rates in every single slot type. The drop in errors is least remarkable for the name and food slots, we believe this is because the challenge in these slots is unknown vocabulary words. Although our framework enriches the dialogue templates with several different slot values, it does so using the values from original set repeatedly. Thus it ends up less helpful for those slots suffering from 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 pricerange: cheap, moderate, expensive).

To support the significance of results on fine-grained slot error types, we use McNemar’s test ($\alpha = 0.01$) upon creating the confusion matrix between our framework and original fine-tuning. The result is $p < \alpha$. Thus we argue that synthetic data fine-tuning shows statistically significant improvements over the original data fine-tuning.

Table 3: The fine-grained errors of original and augmented fine-tuning in restaurant domain classified by slot types.

| Error type             | Original n-shot | Synthetic n-shot |
|------------------------|-----------------|------------------|
| restaurant-food        | 2041            | 1675             |
| restaurant-pricerange  | 1210            | 603              |
| restaurant-name        | 1133            | 1061             |
| restaurant-area        | 853             | 480              |
| restaurant-book day    | 743             | 335              |
| restaurant-book people | 740             | 212              |
| restaurant-book time   | 1119            | 347              |
**Table 4:** 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.

### 4.5.2 Sample Synthetic Dialogues

In this section we showcase two synthetic dialogues generated with our framework, cf. Table 4. These dialogues are generated by merging templates from 2 and 3 different dialogues respectively. The first half of the dialogue on the left includes slots from the restaurant domain, while the second half has slots from the train domain. The dialogue on the right combines slots from domains: train (from two different dialogues) and hotel (from another third dialogue). Although both dialogues seem coherent in shape the sample on the right has a redundancy where the system request the day information after the user already stated it. We found out that this is because of a missing annotation where the train-day slot in the belief state of third turn is missing. These kinds of annotations unavoidable but negligible as well because this in some ways recaptures a misunderstanding by the agent specific to dialogues and since both slots are filled with the same values the coherence of the dialogue is not harmed in a great deal.

### 5 Conclusion

The studies on text augmentation vastly focuses on written text passages implementing sentence or word level transformations to augment original data by generating synthetic samples. None of the studies published so far has attempted to use the embedded information in dialogue states of a task-oriented dialogue which gives important clues about the dialogue function of a turn-pair through its relations with previous and next turn-pairs. In this study, we develop a framework that exploits this idea by making templates by delexicalizing every turn-pair in a dataset and storing them with pointers to previous and next belief states in the dialogue. Using these templates we effectively create augmented dialogue datasets from scratch.

Our experiments using a SOTA model TRADE shows increased performances on DST task for 4 out 5 domains in the MultiWOZ dataset. The only domain not improved, the taxi domain, does not favor even more original data performing equally for 84 (%1 of full data) and 5 dialogues from the original dataset. This exceptional case breaks the motivation of using an augmentation framework. However, considering the significant improvements on other 4 domains we claim that our data augmentation framework consistently improves the performance over the DST task for task-oriented dialogues and it can open doors for many task-oriented dialogue tasks in limited data scenarios, easing the training of data-hungry models in emerging domains.
References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Jūgo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ – A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. 9 2018. URL http://arxiv.org/abs/1810.00278.

Sam Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson. Span-ConvreRT: Few-shot Span Extraction for Dialog with Pretrained Conversational Representations. 2020. URL http://arxiv.org/abs/2005.08866.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. Paraphrase Augmented Task-Oriented Dialog Generation. pages 639–649, 2020. URL http://arxiv.org/abs/2004.07462.

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

Yutai Hou, Yijia Liu, Wanxiang Che, and Bing Xiang. Sequence-to-Sequence Data Augmentation for Dialogue Language Understanding. Technical report, 2018. URL https://github.com/AtmaHou/Seq2SeqDataAugmentationForLU.

Yi Huang, Junlan Feng, Min Hu, Xiaoting Wu, Xiaoyu Du, and Shuo Ma. Meta-Reinforced Multi-Domain State Generator for Dialogue Systems. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7109–7118, 2020. URL https://www.aclweb.org/anthology/2020.acl-main.636.

Shahab Jalalvand, Andrej Ljolje, and Srinivas Bangalore. AUTOMATIC DATA EXPANSION FOR CUSTOMER-CARE SPOKEN LANGUAGE UNDERSTANDING A PREPRINT. Technical report, 2018.

Gakuto Kurata, Bing Xiang, and Bowen Zhou. Labeled Data Generation with Encoder-decoder LSTM for Semantic Slot Filling. 2016. doi: 10.21437/Interspeech.2016-727. URL http://dx.doi.org/10.21437/Interspeech.2016-727.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 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, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1133. URL http://aclweb.org/anthology/P18-1133.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. Coach: A coarse-to-fine approach for cross-domain slot filling, 2020.

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

Jun Quan and Deyi Xiong. Effective Data Augmentation Approaches to End-to-End Task-Oriented Dialogue. Technical report, 2019. URL https://github.com/vsuthichai/paraphraser.

Osman Ramadan, Pawel Budzianowski, and Milica Gašić. 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, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2069. URL http://aclweb.org/anthology/P18-2069.

Abhinav Rastogi, Dilek Hakkani-Tur, and Larry Heck. Scalable Multi-Domain Dialogue State Tracking, 2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017 - Proceedings, 2018-January:561–568, 12 2017. URL http://arxiv.org/abs/1712.10224.

Blaise Thomson and Steve Young. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. Computer Speech and Language, 24(4):562–588, 10 2010. ISSN 08852308. doi: 10.1016/j.csl.2009.07.003.

Zhuoran Wang and Oliver Lemon. A Simple and Generic Belief Tracking Mechanism for the Dialog State Tracking Challenge: On the believability of observed information. Technical report, 2013. URL https://www.aclweb.org/anthology/W13-4067.

Tsung-Hsien Wen, David Vandyke, Nikola Mršić’cmrkš’c, Milica Gašić’cgasi’c, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A Network-based End-to-End Trainable Task-oriented Dialogue System. Technical report, 2017.

Jason D. Williams and Steve Young. Partially observable Markov decision processes for spoken dialog systems. Computer Speech and Language, 21(2):393–422, 4 2007. ISSN 08852308. doi: 10.1016/j.csl.2006.06.008.

Chien-Sheng Wu, Andrea Madotto, Elhsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems, 2019.

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue. Technical report, 2020.

Victor Zhong, Caiming Xiong, and Richard Socher. 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, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1135. URL http://aclweb.org/anthology/P18-1135.