A Method of Automatic Data Construction for Unseen Values in Multi-domain Dialogue State Tracking

Joongbo Shin, Gyeonghun Kim, Hyunjik Jo, Hosung Song, Janghoon Han, Yireun Kim, Stanley Jungkyu Choi
LG AI Research
Seoul, Korea
{jb.shin, ghkayne.kim, hyunjik.jo, hosung.song, janghoon.han, yireun.kim, stanleyjk.choi}@lgresearch.ai

Abstract

The track of `DSTC10 Track2 Task1` aims to benchmark the robustness of dialogue state tracking (DST) models against the gaps between written and spoken conversation. To this end, we develop an approach of automatic data construction to synthesize similar data imitating the validation set of `DSTC10 Track2 Task1`. Among ontology-based, generation-based, and span-based DST models, we decide to use ontology-based DST models since it performs best than others in our synthetic data. In addition, we design hand-crafted rules for utilizing the database for DST, and those rules play an important role in improving the recall of slots. For the final evaluation of `DSTC10 Track2 Task1`, our method achieves 27.03% and 26.7% joint goal accuracy on the validation set and test set, respectively.

Introduction

Developing robust and scalable task-oriented dialogue (TOD) systems has gained increasing attention in both industry and academic communities (Chen et al. 2017). A dialogue system should be able to capture intent from the user utterance and to keep and update this information as the dialog progressed. This task is called dialogue state tracking (DST), which is one of the core components in the TOD systems (Balaraman, Sheikhalishahi, and Magnini 2021). To make the system respond to the user properly, accurate DST is critical for capturing the current state of the conversation. Therefore, it is not too much to say that robust and scalable TOD systems depend on the robustness and scalability of the DST component.

To this end, `DSTC10 Track2 - Task1` proposes a DST task in the extreme situation of spoken conversations without official training data (Kim et al. 2021). During the development period of the challenge, only about 100 dialogues are released for validation. The validation data has similar settings to the MultiWOZ benchmark (Budzianowski et al. 2018), which is one of the popular open-source for multi-domain DST. However, spoken conversations are much noisy than written conversations, and thus much different and difficult to understand. What is worse, databases for TOD systems are different from each other, resulting in encountering unseen named entities. These two challenges, namely *noisy transcripts* and *unseen values*, paralyze any state-of-the-art DST models that are trained on MultiWOZ.

For `DSTC10 Track2 - Task1`, in this work, we first take an approach for automatic data construction that imitates the validation set. We adopt NeuralWOZ (Kim, Chang, and Lee 2021) to generate dialogue and corresponding DST labels using this model-based simulator. To reduce possible errors in the generated data such as non-factual value, we design some hand-crafted rules for generated data normalization. Using our synthetic DST data, we analyze and train several state-of-the-art DST models, and then select the best one. For further improvements, we take an ensemble method and develop additional hand-crafted rules for correcting predicted values. Our DST model achieves 27.02% and 26.79% joint goal accuracy (JGA) on `DSTC10 Track2 - Task1` validation set and test set, respectively.

Related Work

DST Dataset

Because of its importance and difficulty, there have been various data sets for DST such as DSTC2 (Henderson, Thomson, and Williams 2014), WOZ2.0 (Mrkšić et al. 2017), MultiWOZ (Budzianowski et al. 2018), and SGD (Rastogi et al. 2020), to name a few. Amongst them, MultiWOZ is one of the most popular benchmarks for multi-domain dialogue for the DST task due to its natural and realistic conversations Following the Wizard of Oz framework (Kelley 1984), MultiWOZ is collected by Human-Human crowd workers asked to have a conversation between a tourist and a clerk from an information center in a tourist city. It consists of 10k multi-domain dialogues mainly in 5 domains: Attraction, Hotel, Restaurant, Taxi, and Train. MultiWOZ has seen various versions from 1.0 to 2.2 with several error corrections (Ramadan, Budzianowski, and Gasic 2018; Budzianowski et al. 2018; Eric et al. 2020; Zang et al. 2020).

Following the MultiWOZ, `DSTC10 Track2 - Task1` is for the DST task on realistic conversations in 3 domains: *Attraction*, *Hotel*, and *Restaurant*. However, there are obvious differences between MultiWOZ and this challenge. First, as the track proposal said, this track consists of spoken conversations whereas MultiWOZ includes only written conversations. Spoken conversations tend to have extra noises from disfluencies or barge-ins

#Turn	Dialogue	Belief State
#1	USR: I am looking for a **cheap restaurant** in the **centre** of the city.	restaurant-pricerange-cheap;
#2	SYS: There is a cheap chinese restaurant called **Dojo Noodle Bar**.	restaurant-area-centre;
	USR: Yes please , for **8** people at **18:30** on **Thursday**.	restaurant-name-Dojo Noodle Bar;
#3	SYS: Booking has been successful. Anything else you need?	restaurant-book_people-8;
	USR: I am also looking for some **entertainment** close to the restaurant.	restaurant-book_time-18:30;
#4	SYS: Is there any type of attraction you would like me to seach?	restaurant-book_day-Thursday;
	USR: Why do no not try an **architectural** attraction.	attraction-area-centre;
#5	SYS: All **Saints Church** looks good.	attraction-type-architecture
	USR: That's good. Thanks.	
#6	SYS: Thanks.	

Table 1: An example of dialogue state tracking.

and speech recognition errors, making the task more challenging. Second, touristic cities are also different. San Francisco is the target city in `DSTC10 Track2 - Task1` while Cambridge is in MultiWOZ. Therefore, most of named entities such as the name of a hotel and the food of a restaurant in this challenge have never appeared in the MultiWOZ conversations. Those differences require extreme generalization ability of DST models and are our major concerns in this work.

DST models

In DST task, especially research using the MultiWoz dataset, important key points are multi-domain, multi-turn, and detecting slot's value including the special values like *don't care, yes or no, etc*. There are three types of research in DST, one is the ontology-based models such as SUMBT, DST-STAR (Lee, Lee, and Kim 2019; Ye et al. 2021), another is generation-based models such as TARDE, SOM-DST (Wu et al. 2019; Kim et al. 2020), the other is span-based models such as TripPy, SAVN (Heck et al. 2020; Wang, Guo, and Zhu 2020). Ontology-based models utilize the closed vocabulary which is predefined in the ontology. The models select one value from the vocabulary based on the knowledge from the relation between utterances and slot-values (Lee, Lee, and Kim 2019), and from an additional relation between slots for multi-domain and multi-turn (Ye et al. 2021). On the other hand, generation-based models and span-based models detect the open vocabulary values. The generation-based approach generates values of slots in an autoregressive manner. Based on pre-trained language models, modify, or overwrite the values from generated results (Kim et al. 2020). The span-based approach is similar to the Machine Reading Comprehension task, models select the values from the start and end position from the utterance as context. In addition, copy the values from inter-slots, inter-domain (Heck et al. 2020) or normalize the value which detect from spans (Wang, Guo, and Zhu 2020).

Preliminary

In this section, we first provide the definition of the DST task and the details of major challenges in the `DSTC10 Track2 - Task1`.

Problem Definition

Dialogue State Tracking (DST) is a task of estimating and keeping track of the user's goal called *belief state* (or *dialogue state*) as a dialogue progressed. Each turn of the dialogue consists of the system response and user utterance, and the corresponding belief state is represented as a list of $\{$SLOT, VALUE$\}$ pairs. Following the previous works, we use the term *slot* to refer to the concatenation of the domain name and the slot name in order to include both domain and slot information (references). For example, a "*hotel-area*" slot represents the "*area*" of the "*hotel*" domain.

Formally, we denote $D_T = \{(R_1, U_1), \ldots, (R_T, U_T)\}$ as a conversation of T turns, where R_t and U_t are the system response and user utterance at turn t, respectively. Let us to say that we have a set of Q predefined slots $\mathcal{S} = \{S_1, \ldots, S_Q\}$, then the belief state at turn t is defined as $\mathcal{B}_t = \{(S_1, V_1^t), \ldots, (S_Q, V_Q^t)$, where $V_q^t \in \mathcal{V}_q$ denotes the corresponding value of slot S_q. Putting the value spaces of all slots together, we construct an ontology $\mathbf{O} = (S_1, \mathcal{V}_1), \ldots, (S_Q, \mathcal{V}_Q)$, where $\mathcal{V}_q$ is the value space of slot S_q. Then, the DST task is to learn a function $f : D_T \rightarrow \mathcal{B}$. A dialogue example with a corresponding belief state is shown in Table 1.

In the DST task of `DSTC10 Track2`, most formulations are similar to the popular multi-domain DST benchmark, MultiWOZ (Budzianowski et al. 2018). Therefore, in the following section, we mainly explain the challenges of `DSTC10 Track2 - Task1` compared to the MultiWOZ benchmark.

Major Challenges

In this `DSTC10 Track2 - Task1`, there are two major challenges: 1) noisy transcripts and 2) no official training data. Rather than clearly written text, which may have some typo, each dialogue is given as transcripts of spoken conversations by an automatic speech recognition (ASR) system. Following the host, the word error rate of the ASR module is more than 20%, thus user utterances in this track are much noisier than those in MultiWOZ. To compensate for this, the *N*-best list has been offered, but most hypotheses are similar to each other. Besides, we find that manual cleaning has limited effects on the DST accuracy in our preliminary study with the baseline model. Therefore, we take a simple rule-

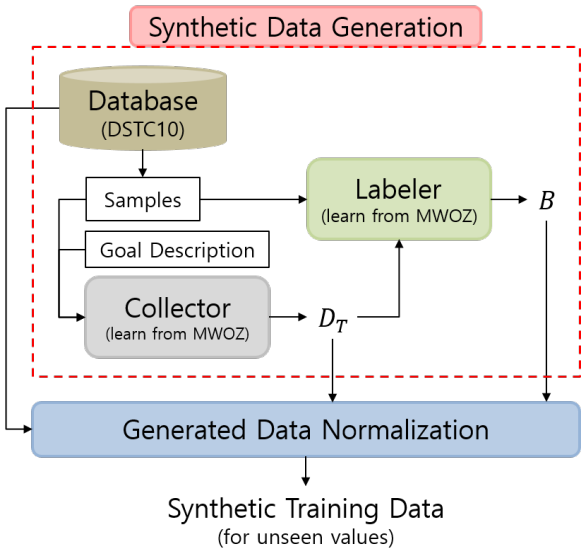

Figure 1: Pipeline of automatic data construction

based text normalization for handling speech recognized errors, and we decide to focus on the second challenge.

Unseen Values During the development period of DSTC10 challenge, database of 3 domains, *Attraction*, *Hotel*, and *Restaurant*, and about 100 unique dialogues are given as a validation set, but there is no official training data. As a baseline, TripPy (Heck et al. 2020) has been suggested to use as a baseline. While it is one of the state-of-the-art DST models in MultiWOZ, showing more than 55% joint goal accuracy (JGA) on the test set, but the performance on the validation set in DSTC10 Track2 - Task1 is frustratingly low, showing barely 0.5% JGA.

We judge that this is because most of the named entities in the validation set are never shown in the training data of MultiWOZ. Since conversations of MultiWOZ is for traveling on Cambridge but those of DSTC10 Track2 - Task1 is for San Francisco, named entities such as names of *attraction/hotel/restaurant* are much differ from each other. Due to this problem of *unseen values*, DST models are easy to not work even though most slots are overlapped.

To fill this gap, in this work, we build training data for this challenge as the most direct approach. For reducing the cost of collecting data, we adopt a simulation-based data construction approach, and this will be detailed in the next section. Note that there has recently been some research on improving DST performances in zero-shot or few-shot settings (Zhao and Eskenazi 2018; Wu et al. 2019; Campagna et al. 2020a; Dingliwal et al. 2021). Making robust DST models is ideal to this problem of course, but most zero-/few-shot DST models have a limitation that using full training data always beats zero-/few-shot settings. Moreover, most researches have focused on unseen domains situation rather than unseen values. For these reasons, we left improving zero-/few-shot DST performances for future work.

Automatic Data Construction

In this section, we describe details of our automatic data construction method. As mentioned above, our major consideration is the problem of the unseen value in multi-domain DST, and our approach to solving this is an automatic data construction. As we have MultiWOZ, one can attempt to use this benchmark might as a template, and to replace values from MultiWOZ with those from the database given in DSTC10 Track2 - Task1. Unfortunately, this simple rule-based modification is prune to raise errors since most of the values are not explicitly reflected in the dialogue. For example, in Table 1 at turn 4, the value of *attraction-type-architecture* is modified as "*architectural*" in the dialogue.

Recently, there have been several synthetic dialogue generation methods (Campagna et al. 2019; 2020b). Among others, we believe that NeuralWOZ (Kim, Chang, and Lee 2021), which is a model-based dialogue simulation for DST, will be a good candidate for generating unseen values. While the original paper aims at the expansion of target domains, namely unseen domains, we find that this model-based approach could reflect a change of the ontology, namely unseen values. Therefore, we build ontology using the database given in the challenge and use them as a source for dialogue simulation. While most synthetic dialogues are well generated and labeled, some of them have errors. As errors will damage the accuracy of DST models, we analyze them and make some hand-crafted rules for deletion, correction, and addition. The overall pipeline of our approach is shown in Figure 1, and details of each component are explained in the following.

Synthetic Data Generation

NeuralWOZ consists of Collector and Labeler, where Collector takes goal description and DB as inputs and then makes multi-turn dialogue, and Labeler takes dialogue, goal description, and DB as inputs and then annotates the dialogue. For other details of NeuralWOZ can be found in the original paper (Kim, Chang, and Lee 2021).

We utilize this approach as a toolkit for converting convert dialogues in Cambridge (MultiWOZ) into those in San Francisco (DSTC10 Track2). Specifically, Collector and Labeler are trained on DST labels with dialogues, goal descriptions, and a database of MultiWOZ. We note that the original usage of NeuralWOz is for unseen domains, but our usage of it is for unseen values. Therefore, we do not exclude any domains during training for the diversity of dialogue.

Before data generation, we exclude Cambridge related database and include San Francisco related database. Also, there are several slot-specific modifications. For example, "*hotel-book day*" and "*restaurant-book day*" include "$Today" and "$Tomorrow" values. In addition, to handle the multi-labeled situation for "*hotel-stars*", we add special vocabulary "at least #" and "more than #", which are directly matched with comma-separated multiple labels. For example, "at least 3" is matched with ["3", "4", "5"].

From these updates and modifications on the database, we let Collector generates 30k dialogues and Labeler annotates them. To make the DST labels compact, we delete

Domain	Slot	#Value
attraction	area	32
	name	128
	type	23
hotel	area	19
	name	200
	pricerange	4
	stars	6+8
	type	6
	book day	9
	book people	11
	book rooms*	2
	book stay	8
restaurant	area	52
	food	121
	name	529
	pricerange	4
	book day	9
	book people	11
	book time	97

Table 2: All slots and the corresponding number of unique values in the given database in `DSTC10 Track2 Task1`. "none" value is excluded in the unique value count. +8 is for handling multiple labels. * is for a newly added slot that is not appeared in the MultiWOZ.

several slots that are included in *Train* and *Taxi* domains. Accordingly, dialogues only about *Train* or *Taxi* domains are excluded. In the (`DSTC10 Track2 Task1`, "*hotel-packing*" and "*hotel-internet*" slots are excluded. Instead, a slot "*hotel-book rooms*", which does not exist in MultiWOZ, is added to the space of belief state. To reflect this slot addition into dialogues, we randomly inject a phrase about the number of rooms only when the dialogue contains information about booking a hotel. In this hand-crafted rule, we assume that tourist requests no more than 3 rooms. The resulting list of slots and the number of unique values for each slot are shown in the Table 2.

Generated Data Normalization

As we utilize model-based data generation from scratch, several dialogues and DST labels have some errors. First, we find that Collector sometimes generates unnatural utterances. As a simple remedy to this problem, we remove samples that contain a word having longer than 32 characters or an utterance having more than 128 words. Also, we notice that Labeler sometimes annotates unintended values that are not in the updated database. Similarly, we remove those unintended values.

Moreover, we find there are non-factual values a lot, which are too many to remove those samples. Therefore, we correct those values into factual values by checking the database automatically. As we modify many values forcibly, some of them could not be reflected in dialogues. To handle this, we design hand-crafted rules for filtering if values are not reflected in the dialogue. For the special value, "dontcare", we cannot take any hand-crafted rule since it can be reflected in the dialogue in various forms. After rule-based filtering, we could increase the value coverage from 87.5% to 94.8%, resulting in reliable data. We also confirm that all named entities in the database appear in the dialogue at least once, which is important to learn end-to-end DST models.

Finally, we make a text normalizer for mimicking ASR transcripts, which is one of the major challenges in this `DSTC10 Track2 Task1`. All characters are converted into lower-cased, and most of the non-alphabets are removed. Especially, we convert all numeric expressions into alphabet sequences, since dialogues in the test set will not have any Arabic number.

Model	Type	Data	JGA
SOM-DST	Generation	ADC w/o GDN	12.4
SOM-DST		ADC	16.2
DST-STAR	Ontology	ADC w/o GDN	17.6
DST-STAR		**ADC**	**20.6**

Table 3: Comparison on Model and training data variation with their performances on validation set. Bold is for the best performance for the single model. ADC denotes our automatic data construction including synthetic data generation (SDG) with generated data normalization (GDN).

TeamID	EntryID	VC	ME	JGA	SA
baseline		no	1	0.0053	0.7056
A10	0	no	1	0.2062	0.8890
A10	1	yes	1	0.2352	0.9022
A10	2	no	3	0.2356	0.8984
A10	3	yes	3	0.2690	0.9046
A10	4	yes	4	0.2702	0.9082

Table 4: Entry modifications and their performances on validation set.

Experiments

Evaluation Metrics

- JGA: Joint goal accuracy measures a proportion of dialog turns for which all the slot values are predicted correctly.
- SA: Slot accuracy measures a proportion of slots for which the model predicts values correctly.

Following the literature and the `DSTC10 Track2 Task1` guideline, we evaluate each DST result mainly using JGA and SA, which are explained above. In addition, Precision, Recall, and F1 of existing values and non-existing (none) values are measured for an in-depth analysis. Note that JGA is the most critical metric for this DST task as the challenge uses this score for the final evaluation.

TeamID	EntryID	JGA	SA	ValueP	ValueR	ValueF1	NoneP	NoneR	NoneF1
baseline		0.0039	0.7052	0.6130	0.3097	0.4115	0.7302	0.9716	0.8338
A10	0	0.1920	0.8850	0.8915	0.7456	0.8120	0.8842	0.9809	0.9300
A10	1	0.2328	0.9016	0.8700	0.8018	0.8345	0.9227	0.9710	0.9462
A10	2	0.2353	0.8973	0.8581	0.7922	0.8238	0.9232	0.9705	0.9463
A10	3	0.2543	0.9046	0.8508	0.8141	0.8320	0.9411	0.9682	0.9544
A10	4	0.2679	0.9079	0.8486	0.8254	0.8368	0.9486	0.9659	0.9571

Table 5: Test results

DST Model Implementation

In this section, our strategies of DST model selection, model ensemble, and value correction are detailed.

Model Selection In general, there are three types of DST models: ontology-based, generation-based, and span-based. Because span-based models such as TripPy (Heck et al. 2020) are not easy to be trained on synthetic data, which has no span information of the values, we only compare ontology-based DST model and generation-based one in this paper. For the ontology-based model, we select a model named DST-STAR, which is the abbreviation of **S**lot self-a**T**tentive dialogue st**A**te t**R**acking (Ye et al. 2021). For the generation-based model, we select a model named SOM-DST, which is the abbreviation of **S**electively **O**verwritting **M**emory for DST (Kim et al. 2020). Both models are ones of state-of-the-arts for each type. As we follow implementations of each official code and use default hyper-parameters, we left the details of model architectures and configurations for the references (Ye et al. 2021; Kim et al. 2020). `BERT-base-uncased` is used in both models as a backbone for a fair comparison.

As shown in the Table 3, DST-STAR models consistently outperform SOM-DST models in the synthetic DST data construction scenario. From qualitative analysis, we find that SOM-DST tends to generate partial values, even with some typos, whereas DST-STAR does not. After this comparison, we finally select DST-STAR trained on our automatic data construction (ADC) as our single model.

Value Correction and Model Ensemble We find that our single model has lower slot recall than slot precision, which means that the model has many missing values to be predicted. In other words, our model tends to predict many values into "None". To increase this value recall rate, we introduce two strategies: value correction (VC) and model Ensemble (ME).

For the value correction, we build some hand-crafted rules for editing predicted values. For one example, booking must be done when the name of a *Hotel* or *Restaurant* has been informed. If names for a *Attraction*, *Hotel*, or *Restaurant* have been predicted as not "none", we search that name in the DB and other related information like "*area*", "*type*" is updated only when those slots are not "none". For the ensemble method, we train DST-STAR model multiple times with different random seeds. To aggregate the predicted values, we use simple majority voting before the value correction. Table 4 shows performances of each entry on validation set for the final submission.

Results and Discussions

Evaluation Scores on the test set are shown in the Table 5, and We confirm that the tendency of the validation set and the test set was the same. We show that our automatic data construction approach is effective for the unseen value problem by showing dramatic improvement compared with the baseline, which is trained on MultiWOZ. The results demonstrate the validity of model ensemble and value correction by improving the value recall consistently while sacrificing the value precision. From those results, we could say that improving the value recall, as well as none precision, is the key to improving JGA in this challenge. In the final evaluation, our best entry achieves the 4th rank on the `DSTC10 Track2 Task1` by obtaining 0.2679 JGA, which is comparable to the 3rd rank (0.2773) and is much better than the 5th rank (0.1821).

Nevertheless, our DST model that is trained on synthetic data seems to lag behind in performances as the single DST-STAR model achieves more than 55% JGA score on MultiWOZ. This suggests that there is much room for improving model-based DST data synthesis for unseen values.s Since we train our DST model on the synthetic dialogue, it is easy to learn a bias of the dialogue simulator. While MultiWOZ has 10k unique dialogues, which are the source for training our dialogue simulator, we generate 30k dialogues repeatedly at random in order to cover all named entities, and it may give rise to learning a bias of the dialogue simulator. Also, we forcibly remove (value) labels that do not appear in the dialogue, which may cause a low recall in our DST model. To remedy this, one could build more sophisticated dialogue simulators or design robust zero-shot or few-shot DST models, which are left for our future works.

Conclusion

For `DSTC10 Track2 - Task1`, in this work, we focus on building a DST model for unseen values with noisy transcripts. By adopting NeuralWOZ, we train a model-based dialogue simulator on the MultiWOZ dataset and generate new dialogues by replacing the database in order to handle unseen values. In order to make synthetic data high quality, we also design data normalization rules, resulting in achieving 20% JGA with a single DST model. We also find that ensemble methods and rule-based value corrections are effective to raise value recall.

References

Balaraman, V.; Sheikhalishahi, S.; and Magnini, B. 2021. Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 239–251.

Budzianowski, P.; Wen, T.-H.; Tseng, B.-H.; Casanueva, I.; Ultes, S.; Ramadan, O.; and Gasic, M. 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*, 5016–5026.

Campagna, G.; Xu, S.; Moradshahi, M.; Socher, R.; and Lam, M. S. 2019. Genie: A generator of natural language semantic parsers for virtual assistant commands. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 394–410.

Campagna, G.; Foryciarz, A.; Moradshahi, M.; and Lam, M. 2020a. 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*, 122–132.

Campagna, G.; Foryciarz, A.; Moradshahi, M.; and Lam, M. 2020b. 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*, 122–132.

Chen, H.; Liu, X.; Yin, D.; and Tang, J. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter* 19(2):25–35.

Dingliwal, S.; Gao, S.; Agarwal, S.; Lin, C.-W.; Chung, T.; and Hakkani-Tur, D. 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*, 1730–1739.

Eric, M.; Goel, R.; Paul, S.; Sethi, A.; Agarwal, S.; Gao, S.; Kumar, A.; Goyal, A. K.; Ku, P.; and Hakkani-Tür, D. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *LREC*.

Heck, M.; van Niekerk, C.; Lubis, N.; Geishauser, C.; Lin, H.-C.; Moresi, M.; and Gasic, M. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *SIGdial*.

Henderson, M.; Thomson, B.; and Williams, J. D. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, 263–272.

Kelley, J. F. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)* 2(1):26–41.

Kim, S.; Yang, S.; Kim, G.; and Lee, S.-W. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 567–582.

Kim, S.; Liu, Y.; Jin, D.; Papangelis, A.; Gopalakrishnan, K.; Hedayatnia, B.; and Hakkani-Tur, D. 2021. "how robust r u?": Evaluating task-oriented dialogue systems on spoken conversations.

Kim, S.; Chang, M.; and Lee, S.-W. 2021. NeuralWOZ: Learning to collect task-oriented dialogue via model-based simulation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 3704–3717. Online: Association for Computational Linguistics.

Lee, H.; Lee, J.; and Kim, T.-Y. 2019. Sumbt: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 5478–5483.

Mrkšić, N.; Séaghdha, D. Ó.; Wen, T.-H.; Thomson, B.; and Young, S. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1777–1788.

Ramadan, O.; Budzianowski, P.; and Gasic, M. 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)*, 432–437.

Rastogi, A.; Zang, X.; Sunkara, S.; Gupta, R.; and Khaitan, P. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 8689–8696.

Wang, Y.; Guo, Y.; and Zhu, S. 2020. Slot attention with value normalization for multi-domain dialogue state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3019–3028. Online: Association for Computational Linguistics.

Wu, C.-S.; Madotto, A.; Hosseini-Asl, E.; Xiong, C.; Socher, R.; and Fung, P. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 808–819.

Ye, F.; Manotumruksa, J.; Zhang, Q.; Li, S.; and Yilmaz, E. 2021. Slot self-attentive dialogue state tracking. In *Proceedings of the Web Conference 2021*, 1598–1608.

Zang, X.; Rastogi, A.; Sunkara, S.; Gupta, R.; Zhang, J.; and Chen, J. 2020. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 109–117.

Zhao, T., and Eskenazi, M. 2018. Zero-shot dialog generation with cross-domain latent actions. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 1–10.