Towards Filling the Gap Between Written and Spoken Dialogues for Multi-Domain Dialogue State Tracking

Taesun Whang[1*], **Jungwoo Lim**[2*], **Dongyub Lee**[3†], **Saebyeok Lee**[1, 2], **Heuiseok Lim**[2‡]

[1] Wisenut Inc., [2] Computer Science and Engineering, Korea University, [3] Naver Corp.
{taesunwhang, saebyeok}@wisenut.co.kr,
{wjddn803, limhseok}@korea.ac.kr, dongyub.lee@navercorp.com

Abstract

The main objective of the task-oriented dialogue system is to identify the intent and needs of the human dialogue. Many existing studies are conducted under the setting of written dialogue, but there always exists a difficulty of coping on real-world spoken dialogues. To this end, DSTC10 challenge organizers propose the task of building robust dialogue state tracking (DST) model on spoken dialogues. On the strength of the existing DST model (i.e., MinTL), this paper suggests integral components to build dialogue state tracker; 1) Data augmentation effectively facilitate the model catch the entities that exist in the evaluation dataset. 2) Levenshtein post-processing aims to prevent the distortion in model prediction caused by the automatic speech recognition errors. To validate the effectiveness of our methods, we evaluate our model on DSTC10 datasets and conduct qualitative analysis by ablating each component of the model. Experimental results show that our model significantly outperforms baselines in all evaluation metrics and took 3rd place in the challenge.

Introduction

The task-oriented dialogue system aims to capture the intents and satisfy the needs in the human dialogue. In the real-world multi-domain dialogue, there are obvious differences between the ways of speaking and writing even for the same context and semantics of the conversations. Moreover, there always exists extra noises from disfluencies or automatic speech recognition (ASR) errors. Along with these challenging situation, DSTC10 proposes the "Knowledge-grounded Task-oriented Dialogue Modeling on Spoken Conversations" track. The objective of the track is to benchmark the robustness of the conversational models while filling the gaps between written and spoken conversations. Task 1 in this track mainly focuses on identifying the state of the given multi-domain dialogues.

The main difficulty of this challenge lies in the fact that the training corpus is not given. Since the validation set is the result of spoken conversation, the most of the dialogue state tracking (DST) datasets available are mainly the written conversation corpus (El Asri et al. 2017; Shah et al. 2018; Wen et al. 2017; Eric et al. 2020). Also, the entities in the training set and those from the evaluation set have significant differences. Under this situation, we set our goal as building a robust and generative dialogue system of open-vocabulary approach to comfortably manage unseen values along with the ASR errors. Moreover, we decide to adopt and implement additional modules to overcome the problem of generating consistent value.

To address these issues, this paper proposes integral components to build applicable models to deal with above real-world errors in spoken conversations. To show the effectiveness of our proposed components, we adopt existing DST model MinTL (Lin et al. 2020), which is an effective transfer learning framework while showing comparable performance with generative pre-trained language models. First, we introduce a highly effective data augmentation strategy to reduce data discrepancy between written and spoken conversations. Since the training dataset is not provided in the challenge, we augment existing DST dataset (e.g., MultiWOZ 2.1 (Eric et al. 2020)) by replacing several names and types of the entities from the given dataset with those of the evaluation dataset. After model training, we then additionally process predicted value to have suitable and consistent dialogue state by exploiting levenshtein post-processing. Lastly, we aggregate the predictions from the differently initialized models by selecting the most predicted value for each slot type and it is taken as a final prediction. Experimental results show that our model outperforms baselines by about 30% in joint goal accuracy and took 3rd place in the challenge.

Related Work

Open Vocabulary-based DST Open vocabulary-based DST is one of the main approaches to traditional dialogues state tracking (DST). Unlike pre-defined ontology-based approach, open vocabulary-based methods (Henderson, Thomson, and Young 2014; Yoshino et al. 2016; Rastogi, Gupta, and Hakkani-Tur 2018) generate slot value at each turn with a generative model such as RNN, LSTM, and GRU. With the advent of pre-trained language models and their remarkable performance (Devlin et al. 2019; Raffel et al. 2020; Lewis et al. 2020), recent studies utilize pre-trained models on DST as well (Chao and Lane 2019; Lin et al. 2020). By exploiting pre-trained language models, the dialogue system does not suffer from task-specific design and extensive hu-

[*] These authors equally contributed to this work.

[†] This work was done at Kakao Corp.

[‡] Corresponding author.

Domain	Augmented Slot Type	Dataset	Dialogues
Restaurant	Name, Area, Day, Food	MultiWOZ 2.1	User: i am looking for a **european** restaurant in **centre** area on **tuesday**. System: **eraina** is a great **european** restaurant in the **centre** of town .
		Augmented Dataset	User: i am looking for a **korean** restaurant in **outer richmond** area on **tomorrow**. System: **um ma son** is a great **korean** restaurant in the **outer richmond** of town .
Hotel	Name, Area, Type, Day	MultiWOZ 2.1	User: I need to book a **hotel** in the **east** that has 4 stars on **monday**. System:There only one **hotel** available in the **east** area. It's called **allenbell**.
		Augmented Dataset	User: I need to book a **hostel** in the **union square** that has 4 stars on **today**. System:There only one **hotel** available in the **union square** area. It's called **adelaide hostel**.
Attraction	Name, Area, Type	MultiWOZ 2.1	User: i am also looking for a **museum** in **west**. System: there are 7 **museum**s in the west . do you have a preference ? User: i have no preference , i just need to know how much the entrance fee is . System: **cambridge and county folk museum** has an entrance fee of 3.50 pounds .
		Augmented Dataset	User: i am also looking for a **amusement park** in **fisherman's wharf**. System: there are 7 **amusement park**s in the **fisherman's wharf**. do you have a preference ? User: i have no preference , i just need to know how much the entrance fee is . System: **7d experience** has an entrance fee of 3.50 pounds .

Table 1: Examples of augmented dataset for each domain in the DSTC10 dataset. All augmented slots except day are extracted from the consistent item. Text in bold indicates augmented slot values.

man annotations. Moreover, the models get benefits from the pre-trained weights and achieve decent performance with a small fraction of the training data.

Handling Automatic Speech Recognition Dialog state tracking models that receive the output of the automatic speech recognition module inevitably face up to ASR errors. Previous studies of handling such errors can be divided into two approaches. One is to consider these errors within the models directly. Henderson, Thomson, and Young (2014); Vodolán, Kadlec, and Kleindienst (2017) explicitly utilize ASR n-best lists as the additional features with extra encoders to find the correct belief state of the dialogue. Schumann and Angkititrakul (2018); Weng et al. (2020) added the layer of correcting ASR errors in training along with the original NLU tasks. Also, Pal et al. (2020) model ASR sequence as graphs and exploit confusion networks with a neural dialogue state tracker. The other is augmenting data simply to the training data for the robust DST model. AdityaTiwari and Polymenakos (2020) leverage an ASR error simulator to inject noise into the error-free text data, and subsequently train the dialog models with the augmented data. Yin et al. (2020) propose a reinforcement learning (RL) based framework for data augmentation that can generate high-quality data to improve the dialog state tracker. Since we aim to build the robust model without latency in the dialog state tracker, we exploit data augmentation to the given training data directly.

Task Description

Multi-domain dialogue state tracking is one of the tasks in DSTC10 track 2; Knowledge-grounded Task-oriented Dialogue Modeling on Spoken Conversations. The main objective of this track is to benchmark the robustness of the conversational models against the gaps between written and spoken conversations (Kim et al. 2021). The dataset is transcribed from the human-to-human dialogues about touristic information for San Francisco. Since it is constructed from the transcription, the data is suffering from ASR errors. The task is evaluated with joint goal accuracy, and the training set is not limited to any of the datasets.

Approach

Problem Formulation

Given a dialogue $\mathcal{D} = \{U_1, R_1, U_2, R_2, ..., U_T, R_T\}$, where U and R is user utterance and system response, respectively, we define dialogue state at each turn $B = \{B_1, B_2, ..., B_T\}$. Slot value for each domain–slot pair is denotes as $B_t(d_i, s_j) = v$, where d is a domain, s is a slot type, and v is a corresponding slot value. The dialogue state tracking model aims to predict dialogue state at each turn B_t for all domain slot pairs, given previous dialogue state B_{t-1} and dialogue context $C_t = \{U_{t-w}, S_{t-w}, ..., S_{t-1}, U_t)\}$, where w is a window size.

MinTL

Lin et al. (2020) introduced the MinTL that leverages generative pre-trained langauge models for multi-domain dialogue state tracking. The main idea of MinTL is to generate dialogue states that needs to be changed (Levenshtein Belief Spans) at each turn. Specifically, we concatenate the previous dialogue states B_{t-1} and dialogue context C_t to build source input. In the case of target sequence S_t, it consists of newly updated slots and each slot is updated based on one of the following conditions.

- **Insertion**: $B_{t-1}(d_i, s_j)$ is a empty value and $B_t(d_i, s_j)$ is newly added at turn t.
- **Deletion**: $B_{t-1}(d_i, s_j)$ has a value and $B_t(d_i, s_j)$ is deleted at turn t (i.e., $B_t(d_i, s_j) = \texttt{None}$).
- **Substitution**: $B_{t-1}(d_i, s_j)$ is replaced with the different value $B_t(d_i, s_j)$ at turn t (Both $B_{t-1}(d_i, s_j)$ and $B_t(d_i, s_j)$ are non-empty values).

Each updates slot is formed as $s_j \oplus B_t(d_i, s_j)$ for each domain and the special token for domain $[d_i]$ is added to the beginning of the very first slot. MinTL model is fine-tuned from generative pre-trained language models, such as

T5 (Raffel et al. 2020) and BART (Lewis et al. 2020). The model is trained by minimizing negative log-likelihood of S_t given B_{t-1} and C_t, which is denoted as,

$$\mathcal{L}oss = -\text{log}p(S_t|B_{t-1}, C_t).$$

Especially, BART is significantly effective when fine-tuned on text generation since it performs well on comprehension tasks also. To obtain the aforementioned advantages, we employ a pre-trained generative language model and the design of Lin et al. (2020) in this study.

Data Augmentation

To build the dataset that covers the entities in the dataset, we augment MultiWOZ dataset by replacing entities of certain slot types in the dialogue. The examples of data augmentation are described in Table 1. In order to reduce the distributional bias of entities, we refer the MultiWOZ labels based on domain and slot types. In other words, we substitute original entities from the MultiWOZ dataset with the corresponding entities from the training set according to the domain and slot types. We randomly choose the entity from the candidates that have the same target domain and slot type to replace. We conduct entity substitution only on the certain slot types as described in Table 1. Along with the entity substitution, only 30 % of the values of slot type `day` are replaced with *today* and *tomorrow* since they are limited to the range of *Monday*, *Tuesday*, *Wednesday*, *Thursday*, *Friday*, *Saturday* and *Sunday* in MultiWOZ dataset.

Levenshtein Post-processing

We introduce method of revising the predicted values depending on the given dataset and database called Lev processing. After obtaining the predicted value from the previous step, we conduct a replacement process utilizing the levenshtein distance (Levenshtein 1992). We first measure the levenshtein distance between predicted value and all the values from the corresponding domain and slot types from the database. After choosing the value that has the lowest score, we also apply word error rate to find the exact matching values to the ground truth.

We also suggest consistent levenshtein post-processing called Lev_c that matches the slot values according to database values of `name`. Once the predicted value of name slot obtained, we additionally find the value of area slots and type slots according to the `name` slots. As we assume that the `name` slot possesses the centralized information of the dialogue, we substitute the previous value with the corresponding value from the database from the matched slot. By exploiting Lev_c, the consistency of the dialogue state increases empirically.

Ensemble

To boost performance, we aggregate the slot value prediction results from several randomly initialized models. All slot values are post-processed (either using Lev or Lev_c) first, and then we select the most predicted value for each slot type as a final prediction. When more than half of the models generates none value (empty slot), none value is taken as a final prediction. On the other hand, if majority of the models generate non-empty values even if the values are slightly different each other, we choose the most predicted value among the non-empty values.

Phase	Training			Evaluation	
Dialogue Type	Written			Spoken	
Dataset	MultiWOZ 2.1			DSTC10	
	Training	Validation	Test	Validation	Test
# dialogues	8434	999	1000	107	783
# turns	56747	7365	7372	936	6588

Table 2: Corpus statistics of dialogue state tracking datasets.

Experiment

Experimental Setup

Dataset To train our model, we adopt clean version of MultiWOZ (Eric et al. 2020; Zhu et al. 2020), which is commonly used benchmark dataset for multi-domain dialogue state tracking task. This dataset is constructed on the basis of written conversations in 7 domains (e.g., restaurant, hotel, police, and taxi). During the training phase, all training, validation, and test sets are used to train the model. For the evaluation, we use DSTC10[*] dataset provided by the challenge organizers. Unlike MultiWOZ, it is annotated from the spoken conversations and covers only 3 domains (i.e., restaurant, hotel, and attraction). To reduce domain discrepancy, we only use dialogue states belonging to these three domains, and those of the remaining domains are not considered during model training. Corpus statistics for both MultiWOZ and DSTC10 datasets are described in Table 2.

Evaluation Metrics We evaluate our methods using several evaluation metrics. 1) Joint goal accuracy is commonly used as the main metric in dialogue state tracking and it is used to rank the participants in the challenge. It gets 1 if all predicted slot-value pairs are exactly same to the ground truth and 0 otherwise at the turn level. 2) Slot accuracy is used to check whether each slot is correctly predicted. 3) Precision, Recall, and F1 score is used for both value and none prediction.

Implementation Details

We implemented our model using PyTorch (Paszke et al. 2019) library. We employed BART-base (Lewis et al. 2020) as a pre-trained backbone model as it showed better results than T5 (Raffel et al. 2020) in our experiments. The batch size is set to 32 and the window size for previous dialogue context is set to 3. For the data augmentation, we replaced slot values with the new entities for every epoch so that the model can learn diverse entities that exist in the database of the evaluation set. The model is trained using Adam optimizer (Kingma and Ba 2014) with the initial learning rate of 2e-5. During training, the ground truth of the current turn is used as the previous state in the next turn. On the other hand, in evaluation, prediction result of the current turn is used as the previous state in the next turn so that the result of the

[*] https://github.com/alexa/alexa-with-dstc10-track2-dataset

	Model	Joint Goal	Slot	Value Prediction			None Prediction		
		Accuracy	Accuracy	Precision	Recall	F1	Precision	Recall	F1
Baselines	TripPy (Heck et al. 2020)	0.53	70.56	56.46	29.93	39.13	74.35	96.10	83.83
	MinTL (Lin et al. 2020)	0.85	75.04	52.24	49.60	50.89	82.54	95.85	88.69
Ours	MinTL + DA	11.54	85.37	71.10	69.90	70.49	94.44	95.44	94.94
	MinTL + DA + Lev	21.26	88.62	79.71	78.36	79.03	94.44	95.44	94.94
	MinTL + DA + Lev$_c$	26.60	87.92	77.77	76.46	77.11	94.45	95.44	94.94
	MinTL + DA + Lev$_c^\dagger$	**30.24**	**90.02**	**81.74**	**81.61**	**81.67**	**95.56**	**95.67**	**95.61**

Table 3: Quantitative results on the DSTC10 validation set. † denotes ensemble model.

front turn is propagated to the that of following turn. For the levenshtein post-processing, we used fuzzywuzzy[†] library to calculate edit distances between generated slot value and database candidates.

Results

Table 3 reports quantitative results on DSTC10 validation set. For the baseline comparison, we compare our models with TripPy (Heck et al. 2020) model, which is the official baseline in the challenge. Also, we report experimental results of the vanilla MinTL (Lin et al. 2020) to show how effective our proposed method is. We perform ablation analyses based on the MinTL model to explore how each method affects the performance improvement. A brief explanation of our proposed methods are as follows.

- **MinTL + DA** aims to reduce data discrepancy between training and evaluation datasets. Slot values related to the information of each item in the database, such as name, area, and type, are replaced to values existing in the evaluation database.
- **MinTL + DA + Lev** finds pre-defined slot values from the database for the slot type in which the value is generated by using levenshtein distance.
- **MinTL + DA + Lev$_c$** finds an item in the database through levenshtein distance for name only. In order to ensure that all generated slot values can be *consistent* for the item, we fill the value of each slot type with the item's pre-defined value.

Compared to the other single models, data augmentation and post-processing based on levenshtein distance significantly improves the joint goal accuracy. Specifically, data augmentation achieves significant improvements in joint goal accuracy from 0.85 to 11.54 and in slot accuracy from 75.04 to 85.37, compared to the MinTL model. Also, levenshtein post-processing (Lev results in an additionoal performance improvement of more than 10% in joint goal accuracy. We also report an ensemble of 15 models (MinTL + DA + Lev$_c$) which are trained with different random initial seed and it achieves the highest performance in all evaluation metrics.

In addition, when comparing the model with Lev and Lev_c, we especially observe that Lev_c significantly increases joint goal accuracy and slightly decreases performance in some metrics related to each slot. Even if the model predicts almost all of the slot values, the joint goal accuracy gets 0 if even one slot is predicted incorrectly. Since the dialogue state of each domain contains values of consistent items, Lev_c consistently replaces them with the searched item information found based on the name. One fatal limitation of this method is that if the searched item is not the correct answer, all replaced values can be wrong so that performance of individual slots is degraded (slot accuracy: -0.7%, value prediction F1: -1.92%).

Rank	Team	Entry	Joint Goal Accuracy	Slot Accuracy	Value F1	None F1
1	A11	1	46.16	94.98	91.15	97.82
2	A01	0	36.05	93.67	89.31	96.72
3	**A07**	**1**	**27.73**	**89.48**	**79.50**	**96.16**
4	A10	4	26.79	90.79	83.68	95.71
5	A09	4	18.21	87.59	79.86	92.33
6	A06	3	16.91	85.95	75.29	93.61
7	A05	3	16.15	86.24	76.55	92.31

Table 4: Official results for test submissions by DSTC10 participants. All rankings are based on the joint goal accuracy metric. Text in bold indicates our model (Team: A07).

Table 4 lists official results for entry submissions by participants. We only report the teams that achieved above 10% in the joint goal accuracy metric. Each team submitted up to 5 prediction results and the result with the highest joint goal accuracy was used for final ranking. We submitted the predictions of MinTL + DA + Lev$_c$ (ensemble) and took 3rd place in the challenge. Even though the winning team achieves remarkable results, it is notable that we obtain significant performance improvement compared to the baseline without using any other ASR corpora and without additional fine-tuning on the DSTC10 validation set.

Qualitative Analysis

Table 5 shows qualitative results on DSTC10 validation set. For the first example, the value of name slot is predicted as *secon one* when we train MinTL using MultiWOZ dataset. Also, the value of the area slot is also predicted as *south* since the values are limited to the *north*, *south*, *east*, and *west* in MultiWOZ After applying data augmentation, the model predicts the name correctly and the area value is converted to *san francisco* which is the value from DSTC10 database on area slots even it is a wrong answer. When we exploit Lev_c, it is shown that the value gains more consistency than that of models utilize Lev. Because the restaurant *um ma son* is in the area of *outer richmond* selling *korean* food, conversion of the values has positive impacts on the results. Since there is an ASR error about the food type in the conversation (i.e.,

[†] https://github.com/seatgeek/fuzzywuzzy

Dialogues	Predictions
User : hi i'm planning a trip to san francisco and i'm looking for recommendation for a u **moderate**ly priced **currying** restaurant in the **outher richmond** area **System**: ok sure let me go and see what i can find. ok so here we do have two options one is called han two kwaan and the second one is **um ma son** which one do you think you might like **User** : the secon one sounds good can you give me their address zip code and phone number please	**MinTL** : [R-name] secon one [R-food] currying [R-area] south [R-pricerange] moderate **MinTL + DA** : [R-name] um ma son [R-food] currying [R-area] san francisco [R-pricerange] moderate **MinTL + DA + Lev** : [R-name] um ma son [R-food] japanese curry [R-area] fisherman's wharf [R-pricerange] moderate **MinTL + DA + Lev$_\mathbf{c}$** : [R-name] um ma son [R-food] korean [R-area] outer richmond [R-pricerange] moderate
	Ground Truth: [R-name] um ma son [R-food] korean [R-area] outer richmond [R-pricerange] moderate
User : e can you help me find a **public market** in the **embarcadero** **System**: certainly sir uhhh i'm showing one location in embarcadero it's called **ferry building market place** **User** : iawesome uh do you know what the address zip code and phone number **System**: definetely so the address is gonna be one ferry building that's b. l. d. g. and their zip code is nine four one one one and their phone number is listed as four one five nine eight three eighty thirty **User** : awesome gool and vitally do you know if uh there's a place i can park bike bike near there **System**: definitely i am showing this location does have bicycle parking yes **User** : thank you	**MinTL** : [A-name] ferry building market place [A-area] centre [A-type] park **MinTL + DA** : [A-name] ferry building market [A-area] embarcadero [A-type] bike rental **MinTL + DA + Lev** : [A-name] ferry building marketplace [A-area] embarcadero [A-type] bike rental **MinTL + DA + Lev$_\mathbf{c}$** : [A-name] ferry building marketplace [A-area] embarcadero [A-type] public market
	Ground Truth: [A-name] ferry building marketplace [A-area] embarcadero [A-type] public market

Table 5: Qualitative results on the DSTC10 validation set. R and A indicate Restaurant and Attraction, respectively.

currying), the model inevitably predicts the food type based on the dialogue. Even Lev method brings the most similar value from the database, it is difficult to predict correct value as the raw value is completely unrelated to the ground truth (i.e., korean). Thus, Lev_c is highly effective in that it brings values for consistent item. Similar consequences can be seen in the second example, especially in the `type` slot. By augmenting training dialogues, MinTL + DA predicts the area as *embarcadero* correctly, not stating *centre*. Moreover, Lev_c also aids to have non-irregular values by switching *bike rental* to *public market*.

Conclusion

In this paper, we focused on reducing data discrepancy between training and evaluation data, and that between written and spoken conversations in multi-domain dialogue state tracking. We proposed a highly effective data augmentation strategy and post-processing method based on levenshtein distance. Experimental results show that our approaches achieve significant improvements in all evaluation metrics. Moreover, we demonstrated how each component affects the outcome through qualitative analysis. For the future work, we plan to train the model for predicting consistent dialogue state for each domain in end-to-end manner rather than post-processing.

Acknowledgement

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 1711117120, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques)

References

AdityaTiwari, L. M.-z.; and Polymenakos, S. M. L. 2020. Data Augmentation for Training Dialog Models Robust to Speech Recognition Errors. *ACL 2020*, 63.

Chao, G.-L.; and Lane, I. 2019. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. *arXiv preprint arXiv:1907.03040*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

El Asri, L.; Schulz, H.; Sarma, S. K.; Zumer, J.; Harris, J.; Fine, E.; Mehrotra, R.; and Suleman, K. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 207–219.

Eric, M.; Goel, R.; Paul, S.; Sethi, A.; Agarwal, S.; Gao, S.; Kumar, A.; Goyal, A.; Ku, P.; and Hakkani-Tur, D. 2020. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines. In *Proceedings of the 12th Language Resources and Evaluation Conference*, 422–428.

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

Henderson, M.; Thomson, B.; and Young, S. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 292–299.

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. arXiv:2109.13489.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Levenshtein, V. I. 1992. On perfect codes in deletion and insertion metric.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880.

Lin, Z.; Madotto, A.; Winata, G. I.; and Fung, P. 2020. MinTL: Minimalist Transfer Learning for Task-Oriented Dialogue Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3391–3405.

Pal, V.; Guillot, F.; Shrivastava, M.; Renders, J.-M.; and Besacier, L. 2020. Modeling asr ambiguity for dialogue state tracking using word confusion networks. *arXiv preprint arXiv:2002.00768.*

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, 8026–8037.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21: 1–67.

Rastogi, A.; Gupta, R.; and Hakkani-Tur, D. 2018. Multi-task Learning for Joint Language Understanding and Dialogue State Tracking. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 376–384.

Schumann, R.; and Angkititrakul, P. 2018. Incorporating asr errors with attention-based, jointly trained rnn for intent detection and slot filling. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6059–6063. IEEE.

Shah, P.; Hakkani-Tür, D.; Tür, G.; Rastogi, A.; Bapna, A.; Nayak, N.; and Heck, L. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871.*

Vodolán, M.; Kadlec, R.; and Kleindienst, J. 2017. Hybrid Dialog State Tracker with ASR Features. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 205–210.

Wen, T.-H.; Vandyke, D.; Mrkšić, N.; Gasic, M.; Barahona, L. M. R.; Su, P.-H.; Ultes, S.; and Young, S. 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*, 438–449.

Weng, Y.; Miryala, S. S.; Khatri, C.; Wang, R.; Zheng, H.; Molino, P.; Namazifar, M.; Papangelis, A.; Williams, H.; Bell, F.; et al. 2020. Joint contextual modeling for asr correction and language understanding. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6349–6353. IEEE.

Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; and Liu, Q. 2020. Dialog state tracking with reinforced data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 9474–9481.

Yoshino, K.; Hiraoka, T.; Neubig, G.; and Nakamura, S. 2016. Dialogue state tracking using long short term memory neural networks. In *Proceedings of Seventh International Workshop on Spoken Dialog Systems*, 1–8.

Zhu, Q.; Zhang, Z.; Fang, Y.; Li, X.; Takanobu, R.; Li, J.; Peng, B.; Gao, J.; Zhu, X.; and Huang, M. 2020. ConvLab-2: An Open-Source Toolkit for Building, Evaluating, and Diagnosing Dialogue Systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 142–149.