Multi-Domain Dialogue State Tracking By Neural-Retrieval Augmentation

Lohith Ravuru[1], Seonghan Ryu[2], Hyungtak Choi[1], Haehun Yang[1], Hyeonmok Ko[1]
[1]Samsung Research, Seoul, South Korea
[2]42dot, Seoul, South Korea
loki.ravuru,ht777.choi,haehun.yang,felix.ko@samsung.com
seonghan.ryu@42dot.ai

Abstract

Dialogue State Tracking (DST) is a very complex task that requires precise understanding and information tracking of multi-domain conversations between users and dialogue systems. Many task-oriented dialogue systems use dialogue state tracking technology to infer users' goals from the history of the conversation. Existing approaches for DST are usually conditioned on previous dialogue states. However, the dependency on previous dialogues makes it very challenging to prevent error propagation to subsequent turns of a dialogue. In this paper, we propose Neural Retrieval Augmentation to alleviate this problem by creating a Neural Index based on dialogue context. Our NRA-DST framework efficiently retrieves dialogue context from the index built using a combination of unstructured dialogue state and structured user/system utterances. We explore a simple pipeline resulting in a retrieval-guided generation approach for training a DST model. Experiments on different retrieval methods for augmentation show that neural retrieval augmentation is the best performing retrieval method for DST. Our evaluations on the large-scale MultiWOZ dataset show that our model outperforms the baseline approaches.

1 Introduction

Dialogue State Tracking (DST) involves analyzing the user's dialogue and previous turn state expressed during the conversation, extracting the user's goal/intent, and representing it in the form of a well-defined set of slots and values (Williams et al., 2016; Henderson, 2015; Williams and Young, 2007; Gao et al., 2018). The release of a large-scale multi-domain conversational data set (MultiWOZ Budzianowski et al., 2018) prompted advances in cross-domain dialogue systems. Figure 1 shows an example from the dataset where the user starts the conversation about reserving a hotel, then requests for booking a taxi, and finally, changes the original hotel reservation. The dialogue state here is defined

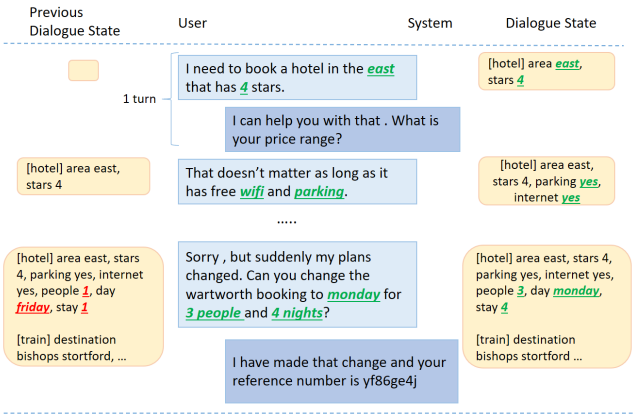

Figure 1: An example from the Large-Scale Multi-Domain Wizard-of-Oz (MutliWOZ) dataset where the user is booking a hotel and a train ticket. The dialogue state is represented as [domain] followed by a list of <slot-value> pairs for that domain. One turn refers to a single user utterance and a single system response. The dialogue state is updated based on the previous dialogue state, the current user utterance and the previous one-turn context.

as list of <slot-value> pairs for each [domain] (e.g., ([hotel] *people* 2 *stay* 5 days) , ([taxi] *departure* Hotel Santa)).

Recent works approach this either by classifying each slot over pre-defined slot-values that are selected from an ontology based on training data (Ma et al., 2019; Li et al., 2020) or first classifying a slot and then detecting the span of text in the original context as value for that slot (Kim et al., 2020; Gao et al., 2019). However, these models are highly dependant on the values in the dataset and the ontology. Another approach to DST is generating the value of a slot or both slot and value using a sequence-to-sequence model (Wu et al., 2019; Le et al., 2020). Papers using large pre-trained models such as GPT2 (Radford et al., 2019) have shown promising results (Budzianowski and Vulić, 2019; Hosseini-Asl et al., 2020). A single generative model can also be used to manage entire dialogue

Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022, pages 169–175
November 20–23, 2022.

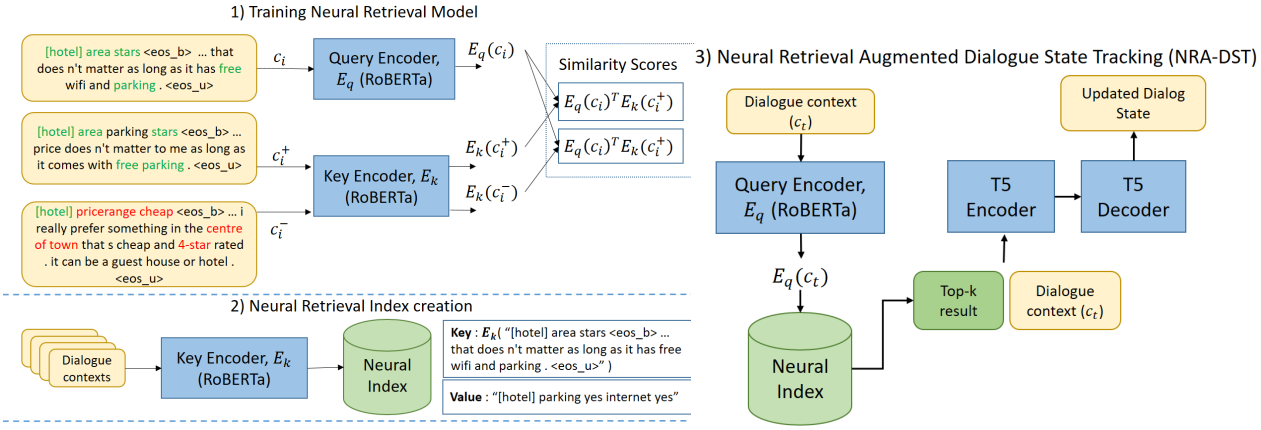

Figure 2: Different steps involved in the NRA-DST approach. The Query Encoder and Key Encoder are trained together. Once trained, Key Encoder is used to create a neural index and Query Encoder is used for retrieving results which are used in finetuning the T5 Model (Raffel et al., 2020), a pretrained Language Model which is used as backbone for our model.

by generating dialogue state, system action, and user response altogether (Lin et al., 2020; Hosseini-Asl et al., 2020). But these models are more prone to error propagation as explained below.

Dialogue State can be considered as a representation of the entire conversation and is used by subsequent modules in resolving system's action and response. Error in the dialogue state propagates not only to these other modules but also to dialogue states of subsequent turns. To analyze this issue, we perform a simple analysis similar to Kim et al. (2020), by replacing the previous dialogue state with ground truth on the state-of-the-art MinTL (Lin et al., 2020) model. As shown in Table 1, using ground truth previous dialogue state in place of the generated previous dialogue state creates a difference of 27% in the prediction of current dialogue state. To bridge the performance gap and reduce error propagation, we propose augmenting retrieved dialogue states of similar dialogue contexts from a pre-computed index.

	Predicted Dialogue State	Actual Dialogue State
MinTL (T5-small)	51.0	78.0
MinTL (T5-base)	51.4	78.3

Table 1: Analysis of Error Propagation in MinTL model.

Large pre-trained models have shown to be very efficient in retrieval-based approaches compared to sparse representations based on TF/IDF, or BM25 (Guu et al., 2020; Lee et al., 2019; Karpukhin et al., 2020). Several works in open domain question answering have augmented retrieval-based results for better response generation (Lewis et al., 2020). However, this is generally done on natural text such as a question or a passage. In Thulke et al. (2021), the retrieval is done using an unstructured dialogue state, but the index is created only from structured paragraph text data.

In this work, we aim to improve DST by leveraging Neural Retrieval-Augmentation on a combination of unstructured dialogue state and structured user/system utterances.

The contributions of our work are as follows:

- We propose an NRA-DST framework that utilizes state-of-the-art neural retrieval methods and integrates it to Dialogue State Tracking for more efficient task-oriented conversations.
- We evaluate our framework on MultiWOZ 2.0 dataset and show that neural retrieval augmentation improves the performance.
- We conduct a comprehensive ablation analysis showing the effectiveness of our proposed framework.

2 Background

In this section, we briefly explain the notations used in further sections. Let us denote the dialogue with t turns as, $D = \{(u_1, r_1), (u_2, r_2), ...(u_t, r_t)\}$, where u_i represents user utterance at i^{th} turn and r_i represents system response at i^{th} turn. Over the course of a dialogue, the goal of DST is to keep track of a dialogue state, $dst = \{(d_1, (s_1, v_1), (s_2, v_2), ..), (d_k, (s_1, v_1)..)\}$ where d_k is the domain, s_i is a slot from the domain, d_k and v_i is the value of s_i. The dialogue context

at turn t is defined as, $c_t = (dst_{t-1}, u_{t-1}, r_{t-1}, u_t)$. In this paper, we formulate dialogue context using only the last turn but this can be extended to multiple previous turns. We formulate the original DST task as predicting the dialogue state from the dialogue context, $dst_t = model(c_t)$.

The concept of Belief Span (Lei et al., 2018) allows dialogue states to be represented as a span of text, enabling the conversion of a classification problem into a generation problem. Lin et al. (2020) builds upon belief spans and defines Levenshtein Belief Span (lev_t) as a minimal editing from previous dialogue state dst_{t-1} to current dialogue state dst_t. For example,

$dst_{t-1} \leftarrow$ [restaurant] food french, price cheap, day Sunday

$dst_t \leftarrow$ [restaurant] food thai, day Sunday, area centre

$$lev_t \leftarrow dst_t - dst_{t-1}$$

$lev_t =$ [restaurant] food thai, price NULL, area centre

We extend the belief spans by creating a neural index and guiding the model with possible Levenshtein spans from the retrieved result. The retrieved topk result contains possible DST updates, $lev_{1..k}$.

$$dst_t = NRADST(lev_{1..k}, c_t) \quad (1)$$

The DST task is now updated as predicting the dialogue state from a combination of retrieved results and dialogue context as in Eq 1. Figure 2 describes the architecture of NRA-DST.

3 Methods

Given a training dataset $D_{train} = \{D_1, D_2, .., D_m\}$, we create a neural index, D_{index} such that we can query the index based on neural representation (latent space representation) of dialogue context c_t, which is a combination of previous dialogue state dst_{t-1} and user/system utterances. Section 3.1 explains the D_{index} creation method in detail. The contents of the D_{index} can be represented as $(E(c_t), lev_t)$, where the key, $E(c_t)$ is the neural representation of dialogue context and the value, lev_t represents the corresponding dialogue state updates. The key idea is that given a dialogue context, we retrieve domains and slots detected in another dialogue with a similar context. Figure 2 shows an example of similar contexts, c_i and c_i^+. The previous dialogue states of both contexts contain the slots named ["area" and "stars"], from the domain named ["hotel"] and the utterances are also similar.

3.1 Neural Dialogue Context Retrieval

For generating efficient Neural Representations, we use a modification of the state-of-the-art Dense Passage Retrieval (DPR) Model (Karpukhin et al., 2020). Similar to the dual-encoder approach proposed in the DPR model, we use two different encoders: Query Encoder (E_q) and Key Encoder (E_k). The DPR model is trained so that the dot-product similarity (Eq 2) is higher for similar dialogue contexts.

$$sim(c_i, c_j) = E_q(c_i)^T E_k(c_j) \quad (2)$$

Training for the similarity metric 2 requires labelling the dataset with positive and negative contexts. For each turn of the dialogue in the training corpus of the original MultiWOZ dataset, we use a customized Algorithm 1 to generate a positive context (c_i^+) and a negative context (c_i^-).

Algorithm 1: Creating Training Data for fine-tuning DPR model.

def *PrepareTrainingInstance*:
- **Input** : Dialogue Context (U_i)
- **Output** : Positive Dialogue Context (U_i^+), Negative Dialogue Context (U_i^-)

Similar Context, $U_{bm25}[100] \leftarrow$ BM25 top100 results from training data;

$Q \leftarrow \{\}$;

$lev_i \leftarrow dst(U_i) - previous\_dst(U_i)$;

foreach *dialogue context* $U_j \in U_{bm25}$ **do**

  $lev_j \leftarrow dst(U_j) - previous\_dst(U_j)$;

  $score \leftarrow slot\_F1(lev_i, lev_j)$;

  $Q.append((score, U_j))$;

end

$sort(Q, key\ a : a[0])$;

$U_i^+, U_i^- \leftarrow Q[0][1], Q[99][1]$;

return U_i^+, U_i^-;

Due to limitations of memory and training time with RoBERTa-base as encoder, we limit the positive and negative contexts to only one context each. We also perform the original DPR model's optimization trick of using in-batch negatives to train effectively. Although we used Algorithm 1 to select only one negative context for a particular training instance, positive contexts from other training

instances in a single training batch are also considered as negative contexts for that instance.

$$L(c_i, c_i^+, ..., c_{i,n}^-) = \\ -log(\frac{e^{sim(c_i, c_i^+)}}{e^{sim(c_i, c_i^+)} + \sum_{k=1}^{n} e^{sim(c_i, c_{i,k}^-)}}) \quad (3)$$

After training the model with the loss function 3, the Key Encoder is used to create the neural index, whereas the Query Encoder is used along with the Dialogue State Tracking model for retrieving the result.

3.2 Generation based Dialogue State Tracking

The retrieval result from Neural Index (lev_{topk}) is appended to the original dialogue context c_t, as described in Eq 1. All sequences are concatenated by using special end-of-sequence (eos) tokens to form a single retrieval-augmented context (c_t^*) and given as input to the T5 (Raffel et al., 2020) encoder.

$$c_t^* \leftarrow lev_1 \ \langle eos\_l1 \rangle \ lev_2 \ \langle eos\_l2 \rangle ... \\ dst_{t-1} \ \langle eos\_b \rangle \ r_{t-1} \ \langle eos\_r \rangle \ u_t \ \langle eos\_u \rangle$$

$$H = Encoder(c_t^*) \quad (4)$$

The T5 decoder model takes as input the encoder hidden states and generates updates to the dialogue state.

$$lev_t = Decoder(H) \quad (5)$$

The loss function used in the Dialogue State Generation model is standard negative loss-likelihood between the ground truth lev_t and generated lev_t. The final dialogue state, dst_t is derived by combining lev_t and dst_{t-1}.

4 Experiments

4.1 Datasets

We evaluate our framework on the Multi-Domain Wizard-of-Oz (MultiWOZ 2.0) (Budzianowski et al., 2018) dataset. The dataset consists of various human-to-human conversations, including tasks from seven different domains (restaurant, train, attraction, hotel, taxi, hospital, police). We used the original dataset split with a training corpus of 8438 dialogues, a validation corpus of 1000 dialogues, and a test corpus of 1000 dialogues.

4.2 Experimental Setup

We implemented our proposed methods on top of the code from MinTL framework (Lin et al., 2020) and Dense-Passage Retrieval model (Karpukhin et al., 2020). For BM25, we use the implementation from Pyserini (pys). We use approximate nearest neighbours with the FAISS library (fai) for performing our retrieval from the neural index. All the hyperparameters used are the default parameters from the baseline implementations.

For our retrieval model, we use RoBERTa (Liu et al., 2019) for Key Encoder, Value Encoder and we use T5-small as the backbone for our DST model. We trained our retrieval model and created the neural index with only the training corpus of the original dataset.

4.3 Metrics

Joint Goal Accuracy measures the accuracy of the generated DST by comparing them to the ground truth DST. The generated slot-value is considered accurate only if it is exactly matching the ground truth slot-value. The accuracy is calculated over each turn for dialogue, and it is averaged over the entire dialogue.

Slot Detection Error is a custom metric that evaluates the benefit of Retrieval Augmentation. It is the error in the ground truth DST and generated DST, but the exact value of the slot is not matched.

4.4 Results

Table 2 describes results on our NRA-DST model compared to other retrieval methods. We compare our model with other generation based baselines DSTQA (Zhou and Small, 2019), NADST (Le et al., 2020), SOM-DST (Kim et al., 2020). We also compare our model with our custom retrieval baselines.

BM25-Retrieval DST Model uses bm25, bag-of-words, retrieval algorithm to create the neural index and retrieve the top-k results.

RoBERTa-Retrieval DST Model uses a pre-trained RoBERTa model directly without any fine-tuning for creating the index.

The decrease in Slot Detection Error and an increase in Joint Goal Accuracy shows that augmenting retrieval results is beneficial for generation-based DST models. We observe that our proposed NRA-DST method outperforms all other retrieval-based models.

Model	Joint Accuracy (↑)	Slot Detection Error(↓)
DSTQA (Zhou and Small, 2019)*	51.44	-
NADST (Le et al., 2020)*	50.52	-
SOMDST (Bert-base) (Kim et al., 2020)*	51.72	-
MinTL (Lin et al., 2020)*†	51.24	-
MinTL†	51.00	12.8
BM25-Retrieval DST†	51.20	12.8
RoBERTa-Retrieval DST†	51.50	12.7
NRA-DST†	**51.90**	**12.5**

Table 2: Results on MultiWOZ 2.0 dataset compared to different baselines. *: results reported by the original paper. †: Uses T5-small model.

5 Ablation Analysis

We analyze the influence of different changes on the results with the following experiments. We try to analyze the importance of previous dialogue state information and delexicalization while creating the neural index and conditioning retrieved results at the encoder or the decoder of our DST model.

5.1 Neural Index

To understand optimal method for neural index preparation, we investigate the effect of using previous dialogue state and delexicalization. Delexicalization is done on the entire dialogue context c_t, which includes removing the slot values from the previous dialogue state and delexicalizing exact slot values from user and system responses. As seen in Table 3, using previous dialogue and delexicalization is very effective.

Previous Dialogue State	Delexicalised Utterances	Joint Accuracy (top1)	Joint Accuracy (top3)
-	-	50.8	50.1
-	✓	50.8	50.9
✓	-	51.3	51.2
✓	✓	**51.9**	**51.2**

Table 3: Ablation comparing different choices of creating neural index and neural retrieval.

In further analysis, we also evaluate our models using top-1 and top-3 retrieved results from the neural index. The results are reported in Table 3. Augmenting top-1 results in better performance than top-3 results. This suggests that augmenting more results is harmful to the performance of the DST models. We reason this as including more retrieved results restricts the number of tokens for dialogue context because of upper limit of 512 tokens for T5 model encoder. To overcome the limit of tokens, we condition the model with the retrieved results on the decoder in the following experiment.

5.2 Augmentation

Model	Joint Accuracy(↑)	Slot Detection Error(↓)
Decoder-NRADST	50.9	12.6
Encoder-NRADST	**51.9**	**12.5**

Table 4: Ablation comparing conditioning retrieval result at encoder and decoder.

Conditioning the retrieval results at the encoder restricts the amount of dialogue context that we can give as input to the model. We experimented with conditioning the retrieved result at the decoder of the T5 model as the actual tokens decoded are much less compared to the dialogue context. Table 4 shows that augmenting at encoder results in the best Joint Accuracy.

6 Conclusions

In this work, we demonstrated that neural retrieval augmentation increases the performance of generation-based DST. We explore a simple pipeline resulting in a retrieval-guided generation approach for DST. Moreover, our experiments and ablation studies indicate that neural retrieval can efficiently retrieve a combination of unstructured data (dialogue state) and structured data (user/system utterances). As a result, we improve the performance of the baseline approach on a large-scale multi-domain dataset, MultiWOZ 2.0. In future work, we will investigate the end-to-end training of our NRA-DST framework.

References

https://github.com/castorini/pyserini.

https://github.com/facebookresearch/faiss.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong. Association for Computational Linguistics.

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

Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 607–613, Brussels, Belgium. Association for Computational Linguistics.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of Special Interest Group on Discourse and Dialogue*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Matthew Henderson. 2015. Machine learning for dialog state tracking: A review. In *Proceedings of The First International Workshop on Machine Learning in Spoken Language Processing*.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582, Online. Association for Computational Linguistics.

Hung Le, Richard Socher, and Steven C.H. Hoi. 2020. Non-autoregressive dialog state tracking. In *International Conference on Learning Representations*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *CoRR*, abs/1906.00300.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Melbourne, Australia. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401.

Jieyu Li, Su Zhu, and Kai Yu. 2020. A hierarchical tracker for multi-domain dialogue state tracking. In *Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yue Ma, Zengfeng Zeng, Dawei Zhu, Xuan Li, Yiying Yang, Xiaoyuan Yao, Kaijie Zhou, and Jianping Shen. 2019. An end-to-end dialogue state tracking system with machine reading comprehension and wide & deep classification. In *Proceedings of The 34th AAAI Conference on Artificial Intelligence - DSTC 8 Workshop*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

David Thulke, Nico Daheim, Christian Dugast, and Hermann Ney. 2021. Efficient retrieval augmented generation from unstructured knowledge for task-oriented dialog. *CoRR*, abs/2102.04643.

Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue Discourse*, 7:4–33.

Jason D. Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Comput. Speech Lang.*, 21(2):393422.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. *CoRR*, abs/1905.08743.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *CoRR*, abs/1911.06192.