Effective Sequence-to-Sequence Dialogue State Tracking

Jeffrey Zhao, Mahdis Mahdieh, Ye Zhang, Yuan Cao, Yonghui Wu
Google Research
{jeffreyzhao,mahdis,yezhan,yuancao,yonghui}@google.com

Abstract

Sequence-to-sequence models have been applied to a wide variety of NLP tasks, but how to properly use them for dialogue state tracking has not been systematically investigated. In this paper, we study this problem from the perspectives of pre-training objectives as well as the formats of context representations. We demonstrate that the choice of pre-training objective makes a significant difference to the state tracking quality. In particular, we find that masked span prediction is more effective than auto-regressive language modeling. We also explore using Pegasus, a span prediction-based pre-training objective for text summarization, for the state tracking model. We found that pre-training for the seemingly distant summarization task works surprisingly well for dialogue state tracking. In addition, we found that while recurrent state context representation works also reasonably well, the model may have a hard time recovering from earlier mistakes. We conducted experiments on the MultiWOZ 2.1-2.4, WOZ 2.0, and DSTC2 datasets with consistent observations.

1 Introduction

Sequence-to-sequence (Seq2Seq) modeling (Sutskever et al., 2014) is one of the most widely adopted generative framework for a multitude of NLP tasks. While it has also been applied for task-oriented dialogue modeling (Wen et al., 2018; Lei et al., 2018; Rongali et al., 2020; Feng et al., 2020), how to best setup Seq2Seq models on this task remains an understudied topic. In this paper, we investigate this problem from two perspectives: Pre-training objectives and dialogue context representation, and we focus on the dialogue state tracking (DST) task.

The flexibility of the Seq2Seq model allows us to adopt and compare pre-training strategies for other NLP tasks sharing the same architecture. Specifically, we first experimented with different pre-training setups of T5 (Raffel et al., 2020) which have been shown to be effective for generic language understanding. Additionally, as an exploratory effort, we applied Pegasus (Zhang et al., 2020b), a pre-training procedure designed for text summarization, to the task of DST.

Additionally, to investigate how different dialogue context representations affect Seq2Seq performance, we compare two versions of all models: one accepts full conversation history as context, and one that feeds the previously predicted states recurrently as summary of context.

We conduct systematic experiments on the MultiWOZ (Budzianowski et al., 2018) benchmark. For fair comparison with existing approaches, we report results on MultiWOZ 2.1-2.4 (Eric et al., 2019; Zang et al., 2020; Han et al., 2020; Ye et al., 2021), all 4 variations of the benchmark proposed to date. In addition, we report results on the WOZ 2.0 (Wen et al., 2016) and DSTC2 (Henderson et al., 2014) datasets. Our findings can be summarized as follows:

1. Pre-training procedures involving masked span prediction work consistently better than auto-regressive language modeling objectives.
2. Pre-training for text summarization works surprisingly well for DST, despite it being a seemingly irrelevant task.
3. Recurrent models work reasonably well by including previously predicted states and constant length dialogue history. However they may suffer from the problem of not being able to recover from early mistakes.

2 Methods

2.1 Models

We directly apply the Seq2Seq model to the problem of state tracking, where both the encoder and decoder are Transformers (Vaswani et al., 2017).

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7486–7493
November 7–11, 2021.

The inputs to the encoder are dialogue contexts, and the decoder generates a sequence of strings of the format `slot1=value1,slot2=value2,...` describing the predicted states conditioned on the given context. Depending on how we represent the dialogue context, we consider two variations of the model:

1. **Full-history model**: The most straightforward way of preparing the context is simply to concatenate turns from the entire history as inputs to the encoder, which ensures the model to have full access to the raw information required to predict the current state. This setup is also adopted by several generative dialogue models such as SimpleTOD (Hosseini-Asl et al., 2020), Seq2Seq-DU (Feng et al., 2020) and SOLOIST (Peng et al., 2021). A potential drawback of the full-history model is that it may become increasingly inefficient as a conversation unfolds and the input length grows.

2. **Recurrent-state model**: An alternative approach is to include just the N recent turns in the conversation history, and replace turns from 1 to $T-N$ with dialogue states up to $T-N$ (where T is the current turn index). That is, the inputs to encoder have the format $[\texttt{states}(turn_{1,\ldots,T-N}), turn_{T-N+1,\ldots,T}]$. States provide a summarization of the conversation semantics. By consolidating remote histories into states we not only reduce the context lengths, but also discard information not immediately related to the purpose of state tracking. Similar setup has also been considered by generative models including GPT-2 (Budzianowski and Vulić, 2019) and Sequicity (Lei et al., 2018), although in their cases only the last turn has been considered ($N = 1$).

An example of the input and output formats for both models is given in Appendix A.1.

2.2 Pre-training

Pre-training followed by task-specific fine-tuning is becoming a standard paradigm for contemporary NLP model training. Existing pre-training objectives mainly fall into two categories: masked span prediction (where the span length can be 1 corresponding to word prediction) and auto-regressive prediction. Objectives like BERT MLM (Devlin et al., 2019) and the denoising setup in T5 (Raffel et al., 2020) belong to the former category, while GPT (Radford et al., 2019; Brown et al., 2020) and the prefix LM setup in T5 fall into the latter.

For generative dialogue modeling, both pre-training styles have been considered. For example, Seq2Seq-DU (Feng et al., 2020) adopted a BERT-pre-trained encoder, while SimpleTOD (Hosseini-Asl et al., 2020) and SOLOIST (Peng et al., 2021) are based on the GPT-2 auto-regressive prediction procedure. Nevertheless, it remains unclear which style is more effective for dialogue understanding. To study this problem, we compare span prediction and auto-regressive language modeling (ARLM) by pre-training the encoder and decoder simultaneously using the denoising and prefix LM objectives from T5. To compare the relative effectiveness of different pre-training styles, we consider 3 setups: 1) Pre-training the model with span prediction only; 2) Continuing the pre-training of models from setup (1) with prefix LM; 3) Pre-training the model only with prefix LM only.

While T5 pre-training has demonstrated its effectiveness for generic language understanding tasks such as the GLUE and SuperGLUE benchmarks, we are curious as to which procedures are biased towards the downstream DST task. While it can be difficult to define an objective that applies immediately to DST, we consider a surrogate pre-training for a seemingly remote task: Summarization. To properly summarize a large chunk of text requires the model to be able to extract key semantics out of a clutter of inputs, which to some extent shares a similar problem structure as DST.

Following this intuition, we choose Pegasus (Zhang et al., 2020b), a strong pre-training objective developed for summarization based on Seq2Seq, as an alternative for comparison. In brief, Pegasus defines a self-supervised objective named Gap Sentence Generation (GSG), which identifies potentially important sentences in a paragraph according to some heuristics (for example, the top-m sentences with the highest ROUGE score with respect to the remaining ones), masks them out, and forces the decoder to predict these pivoting sentences. A critical difference between Pegasus and other span prediction objectives is that masked spans are carefully identified instead of randomized. This principled operation positions the model to work specifically well for the downstream task of summarization.

3 Experiments

3.1 General Setup

Our models are built with the open-source framework Lingvo (Shen et al., 2019)[1]. Each encoder and decoder has 12 Transformer layers, 8 attention head's and embedding dimension 768. Our models are trained with 16 TPUv3 chips (Jouppi et al., 2017). We use the memory-efficient Adafactor (Shazeer and Stern, 2018) as the optimizer, with learning rate 0.01 and inverse squared root decay schedule. We use the default SentencePiece model provided by T5[2] with vocabulary size 32k. For the pre-training procedure, we strictly follow the setups and procedures described in (Zhang et al., 2020b) and (Raffel et al., 2020). For decoding, we use beam search with size 5. We also enabled label smoothing with uncertainty 0.1 during training.

3.2 Datasets

We conduct our experiments on the MultiWOZ (Budzianowski et al., 2018) benchmark. The original MultiWOZ dataset, released in 2018, was known to contain substantial annotation errors. Continuous efforts have been made in recent years to clean up and refine the annotations, resulting in 4 variations of the dataset (2.1-2.4, Eric et al. (2019); Zang et al. (2020); Han et al. (2020); Ye et al. (2021)). The existence of multiple versions of the same benchmark, as well as ad-hoc pre- and post-processing procedures[3] adopted by different research groups make it difficult to compare results fairly. We therefore report results on all of MultiWOZ 2.1-2.4[4], *without* any pre- or post-processing of the original data. We use Joint-Goal-Accuracy (JGA) as the metric for all experiments.

In addition to the MultiWOZ datasets, we also report results on the WOZ 2.0[5] (Wen et al., 2016) and DSTC2[6] (Henderson et al., 2014) datasets. While these datasets are much smaller in both ontology and number of examples when compared to MultiWOZ, they provide additional evidence for the conclusions we make in this paper.

We compare our results with a set of strong baselines: TRADE (Wu et al., 2019), SUMBT (Lee et al., 2019), DS-DST (Zhang et al., 2020a), Seq2Seq-DU (Feng et al., 2020), SOM-DST (Kim et al., 2020), Transformer-DST (Zeng and Nie, 2021), TripPy (Heck et al., 2020), SAVN (Wang et al., 2020), SimpleTOD (Hosseini-Asl et al., 2020), StateNet (Ren et al., 2018), GLAD (Zhong et al., 2018), GCE (Nouri and Hosseini-Asl, 2018), and Neural Belief Tracker (Mrksic et al., 2016). To be consistent with our approach, when both open- and closed-vocabulary setups are available, we only compare with the open-vocabulary setup.

Note that on DSTC2, unlike other methods which combines the n-best speech recognition hypotheses as inputs, we make use of only the top 1-best hypothesis for simplicity, although the combination of n-best hypotheses could potentially further improve DST quality.

3.3 Results

We first report the MultiWOZ JGA scores achieved by the full-history models in Table 1, in which "span" and "ARLM" indicate masked span prediction and auto-regressive language modeling for pre-training respectively, and "span+ARLM" means pre-training with "span" followed by "ARLM". In addition, WOZ and DSTC2 JGA scores are reported in Table 2.

From Tables 1 and 2 we make the following observations:

1. Pre-training procedures with masked span prediction involved ("span", "span+ARLM") consistently performed better than using "ARLM" alone. This is true even if "span" is continued by "ARLM", and this result is seen in not just MultiWOZ 2.1-2.4 but also WOZ 2.0 and DSTC2.

2. Pegasus pre-training works almost equally well or better than the T5 pretraining, indicating that some features can be shared and transferred between the two tasks. Again, this observation is consistent across all benchmarks. This also corroborates conclusion (1) above in that span prediction objectives are more effective for DST.

[1] https://github.com/tensorflow/lingvo

[2] https://github.com/google-research/text-to-text-transfer-transformer

[3] For example on MultiWOZ 2.1, some well-known works including TRADE (Wu et al., 2019), TripPy (Heck et al., 2020) and SimpleTOD (Hosseini-Asl et al., 2020) applied different data processing procedures, making the results incomparable.

[4] MultiWOZ datasets retrieved from: https://github.com/budzianowski/multiwoz (2.1, 2.2), https://github.com/lexmen318/MultiWOZ-coref (2.3), https://github.com/smartyfh/MultiWOZ2.4 (2.4)

[5] WOZ 2.0 dataset retrieved from https://github.com/nmrksic/neural-belief-tracker/tree/master/data/woz

[6] DSTC2 dataset retrieved from https://github.com/matthen/dstc

Model	MultiWOZ 2.1★	2.2	2.3	2.4
TRADE	45.6	45.4	49.2	55.1
SUMBT	49.2	49.7	52.9	61.9
DS-DST	51.2	51.7	-	-
Seq2Seq-DU	-	**54.4**	-	-
Transformer-DST	**55.35**	-	-	-
SOM-DST	51.2	-	55.5	**66.8**
TripPy	55.3	-	**63.0**	59.6
SAVN	54.5	-	58.0	60.1
SimpleTOD▲	50.3/**55.7**	-	51.3	-
Pegasus	**54.4**	56.6	**60.2**	66.6
T5 (span)	52.8	**57.6**	59.3	**67.1**
T5 (span+ARLM)	53.1	57.1	59.9	65.6
T5 (ARLM)	52.5	56.1	58.9	63.0
No pre-train	25.8	26.1	28.1	26.7

Table 1: JGA comparison on MultiWOZ 2.1-2.4 with the full history model. ★: For 2.1, baseline methods adopted different and incomparable data clean-up procedures, but we used the original data and did not do any pre- or post-processing for convenient and fair future comparisons. ▲: SimpleTOD results are cited from the 2.3 website https://github.com/lexmen318/MultiWOZ-coref, in which two numbers are reported for 2.1 (one produced by the 2.3 author, the other by the original SimpleTOD paper). "-" indicates no public number is available. Best results given by existing and our models are marked in bold.

Model	WOZ 2.0	DSTC2
SUMBT	**91.0**	-
StateNet-PSI	88.9	**75.5**
GLAD	88.1	74.5
GCE	88.5	-
Neural Belief Tracker: NBT-DNN	84.4	72.6
Neural Belief Tracker: NBT-CNN	84.2	73.4
Pegasus	**91.0**	**73.6**
T5 (span)	**91.0**	**73.6**
T5 (span+ARLM)	**91.0**	73.5
T5 (ARLM)	89.5	73.3
No pre-train	64.5	50.1

Table 2: JGA comparison for WOZ 2.0 and DSTC2 datasets on the full history model. Note that our DSTC2 JGAs are likely underreported. While other models use the n-best predictions to evaluate, we only used the single best prediction.

3. Without pre-training, model quality drops miserably, as expected.

3.4 Recurrent Results

For the recurrent-state model, we report results for the Pegasus pre-trained model on MultiWOZ 2.1-2.4 in Table 3, with $N = 1, 2, 3$ (number of recent history turns). Each turn contains a pair of user and agent utterance. Our observations on models pre-trained with T5 are similar. The results show that while the recurrent-state models achieved reasonably good JGA on all data sets, they are nevertheless worse than the full-history model, despite the fact that the representation of context is more concise for the recurrent model. What is more, the choice of N can make a big difference to the model quality.

Model	2.1★	2.2	2.3	2.4
Pegasus (full-history)	54.4	56.6	60.2	66.6
Pegasus (1-turn history)	52.4	53.9	**59.5**	57.0
Pegasus (2-turn history)	**52.7**	**56.2**	59.2	59.3
Pegasus (3-turn history)	52.3	55.9	58.4	**60.0**

Table 3: JGA of the recurrent model pre-trained with Pegasus. ★ has the same meaning as in Table 1.

A closer look at failed examples produced by the model reveals that the main reason why the recurrent context representation achieved worse results is that they had a hard time recovering from prediction mistakes made at earlier turns. Since previously predicted states are feedback to the model inputs for future predictions, as long as the model made a mistake at earlier turns, this wrong prediction will be carried over as future inputs, causing the model to make consecutive prediction errors. We therefore believe that for the DST task, it may still be important to provide the model full access to dialogue history, so that it can learn to correct its predictions once a mistake was made in the past.

3.5 Remarks

From results enumerated in Sec. 3.3, one will see wildly varying scores across MultiWOZ 2.1-2.4, despite the fact that each dataset evolved from the same benchmark. This poses a concerning question of whether existing approaches can generalize well across different setups and benchmarks. For example, TripPy performed remarkably well on 2.1 and 2.3 (55.3%, 63.0%), but dropped to 59.6% on 2.4 (which claims to be the "cleanest" version of MultiWOZ). SOM-DST on the other hand, underperformed on 2.1 and 2.3 but achieved a strong

result on 2.4.

We therefore suggest researchers working on the MultiWOZ benchmark report results on multiple version of the data with consistent or no data processing steps, to provide the community a more faithful assessment of the quality of their approaches.

4 Related Work

Generative sequential models have been applied for task-oriented dialogue problems in several ways. (Budzianowski and Vulić, 2019; Hosseini-Asl et al., 2020; Peng et al., 2021) adopted GPT-2, a unidirectional pretrained Transformer LM, as backbones for the generation of states, actions and responses. Under the framework of Seq2Seq, perhaps most similar to our work is (Feng et al., 2020), which adopts a Transformer encoder-decoder architecture, with the encoder pre-trained with BERT which is also used to encode schema. Besides, (Wen et al., 2018) is an early example that uses encoder outputs as state representation, merged with KB representation for the decoder to generate responses; (Lei et al., 2018) proposes a simplistic two stage CopyNet on top of Seq2Seq model to enable word copying from input sequences; (Chen et al., 2020) proposes a hierarchical Seq2Seq model for coarse-to-fine DST; (Zeng and Nie, 2021) proposes a "flat" encoder-decoder structure which reuses a BERT encoder for the function of a decoder with hidden layer states reused.

In terms of pre-training, BERT and GPT are still the most commonly used techniques (Zaib et al., 2021; Zhang et al., 2020c). Various pre-training methods developed for dialogue-specific problems have also been developed. (Zhang et al., 2021) uses dialogue specific datasets for pre-training and fine-tuning. (Mehri et al., 2019) studies 4 ways of pre-training aiming at better capturing discourse-level dependencies for multi-turn dialogues; (Li et al., 2020) proposed a contrastive pre-training loss to capture important qualities of dialogues; (Bao et al., 2021) proposed a curriculum pre-training procedure for response generation, subsuming open-domain, knowledge-grounded, task-oriented dialogue applications. (Liu et al., 2021) factorize the generative dialogue model according to the noisy channel model, pre-training each component separately.

5 Conclusion

We studied the problem of how to perform the DST task with Seq2Seq models effectively from the perspective of pre-training and context representation. We demonstrated that Seq2Seq pre-training objectives involving masked span prediction are more preferred than auto-regressive predictions for dialogue understanding. This observation further generalizes to the adoption of Pegasus, a span prediction objective for summarization, which works surprisingly well on DST tasks. We also find that recurrent state representation for dialogue context can work reasonably well.

References

Siqi Bao, Bingjin Chen, Huang He, Xin Tian, Han Zhou, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, and Yingzhan Lin. 2021. A unified pre-training framework for conversational ai.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pre-trained 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.

Zhi Chen, Lu Chen, Zihan Xu, Yanbin Zhao, Su Zhu, and Kai Yu. 2020. Credit: Coarse-to-fine sequence generation for dialogue state tracking.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines.

Yue Feng, Yang Wang, and Hang Li. 2020. A sequence-to-sequence approach to dialogue state tracking.

Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Wei Peng, and Minlie Huang. 2020. Multiwoz 2.3: A multi-domain task-oriented dataset enhanced with annotation corrections and co-reference annotation.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 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*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

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

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

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, and al. et. 2017. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, ISCA '17, page 1–12, New York, NY, USA. Association for Computing Machinery.

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.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

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.

Junlong Li, Zhuosheng Zhang, Hai Zhao, Xi Zhou, and Xiang Zhou. 2020. Task-specific objectives of pre-trained language models for dialogue adaptation.

Qi Liu, Lei Yu, Laura Rimell, and Phil Blunsom. 2021. Pretraining the noisy channel model for task-oriented dialogue.

Shikib Mehri, Evgeniia Razumovskaia, Tiancheng Zhao, and Maxine Eskenazi. 2019. Pretraining methods for dialog context representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3836–3845, Florence, Italy. Association for Computational Linguistics.

Nikola Mrksic, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve J. Young. 2016. Neural belief tracker: Data-driven dialogue state tracking. *CoRR*, abs/1606.03777.

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking model. *CoRR*, abs/1812.00899.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching.

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.

Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. 2018. Towards universal dialogue state tracking. *CoRR*, abs/1810.09587.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. In *Proceedings of The Web Conference 2020*, WWW '20, page 2962–2968, New York, NY, USA. Association for Computing Machinery.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, Mia X. Chen, Ye Jia, Anjuli Kannan, Tara Sainath, Yuan Cao, Chung-Cheng Chiu, Yanzhang He, and Jan Chorowski et al. 2019. Lingvo: a modular and scalable framework for sequence-to-sequence modeling.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Yexiang Wang, Yi Guo, and Siqi Zhu. 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)*, pages 3019–3028, Online. Association for Computational Linguistics.

Haoyang Wen, Yijia Liu, Wanxiang Che, Libo Qin, and Ting Liu. 2018. Sequence-to-sequence learning for task-oriented dialogue with dialogue state representation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3781–3792, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve J. Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *CoRR*, abs/1604.04562.

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

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2021. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation.

Munazza Zaib, Quan Z. Sheng, and Wei Emma Zhang. 2021. A short survey of pre-trained language models for conversational ai-a newage in nlp.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 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 (2020)*, pages 109–117.

Yan Zeng and Jian-Yun Nie. 2021. Jointly optimizing state operation prediction and value generation for dialogue state tracking.

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S. Yu, Richard Socher, and Caiming Xiong. 2020a. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020b. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Ye Zhang, Yuan Cao, Mahdis Mahdieh, Jeffrey Zhao, and Yonghui Wu. 2021. Improving longer-range dialogue state tracking. *arXiv preprint arXiv:2103.00109*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020c. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive dialogue state tracker. *CoRR*, abs/1805.09655.

A Appendix

A.1 Dialog Example

Dialog examples are formatted to the following input and output sequences for the models presented in this paper. An example input sequence for a full-history model:

```
user: I need to find a spot on a train
on wednesday, can you help me find one?
agent: yes I can. where are you going
and what time would like to arrive or
depart? user: i'm leaving from london
kings cross and going to cambridge. i'd
like to leave after 14:30 on wednesday.
agent: where would you be departing
from? user: i am looking to depart
from broxbourne.
```

The target output sequence would be

```
train-day = wednesday; train-departure =
london kings cross; train-destination =
cambridge; train-leaveat = 14:30
```

As a recurrent example, we would remove the older turns of the conversation and replace them with the relevant states. For example, the 2-turn recurrent input sequence for this example would be

```
<state> train-day = wednesday
<utterance> agent:  yes I can.  where
are you going and what time would like
to arrive or depart?  user:  i'm leaving
from london kings cross and going to
cambridge.  i'd like to leave after
14:30 on wednesday.  agent:  where would
you be departing from?  user:  i am
looking to depart from broxbourne.
```

The 1-turn recurrent input sequence would be

```
<state> train-day = wednesday;
train-departure = london kings
cross; train-destination = cambridge;
train-leaveat = 14:30 <utterance> agent:
where would you be departing from?  user:
i am looking to depart from broxbourne.
```

Note that the target output sequence remains the same for the recurrent input sequences. That is, the model is expected to carry over the predictions from previous states into the current state. Empirically, we found this approach works better than only predicting the new states at each turn.