When does Further Pre-training MLM Help? An Empirical Study on Task-Oriented Dialog Pre-training

Qi Zhu[1], **Yuxian Gu**[1], **Lingxiao Luo**[1], **Bing Li**[1],
Cheng Li[2], **Wei Peng**[2], **Xiaoyan Zhu**[1], **Minlie Huang**[1*]
[1]CoAI Group, DCST, IAI, BNRIST, Tsinghua University, Beijing, China
[2]Artificial Intelligence Application Research Center, Huawei Technologies, Shenzhen, China
zhu-q18@mails.tsinghua.edu.cn, aihuang@tsinghua.edu.cn

Abstract

Further pre-training language models on in-domain data (domain-adaptive pre-training, DAPT) or task-relevant data (task-adaptive pre-training, TAPT) before fine-tuning has been shown to improve downstream tasks' performances. However, in task-oriented dialog modeling, we observe that further pre-training MLM does not always boost the performance on a downstream task. We find that DAPT is beneficial in the low-resource setting, but as the fine-tuning data size grows, DAPT becomes less beneficial or even useless, and scaling the size of DAPT data does not help. Through Representational Similarity Analysis, we conclude that more data for fine-tuning yields greater change of the model's representations and thus reduces the influence of initialization.[1]

1 Introduction

Pre-trained models such as BERT (Devlin et al., 2019) and GPT2 (Radford et al., 2019) have been used in a wide range of NLP tasks and achieved superior performance. These models usually follow the *pre-train* and *fine-tune* paradigm, which adopts unsupervised pre-training on large-scale corpora and supervised fine-tuning for downstream task adaption. However, the pre-training corpora are in the general domain, while the data of downstream tasks fall in more task-specific domains.

To bridge the data distribution gap, further pre-training has been applied and shows consistent improvements (Sun et al., 2019). According to the training data used in this process, Gururangan et al. (2020) termed *domain-adaptive pre-training* (DAPT), which uses the data in the same domain of the target task and *task-adaptive pre-training* (TAPT), which uses much less unlabeled training data from the target task than DAPT. They found that DAPT masked LM leads to performance gains under both high- and low-resource settings and TAPT is beneficial with or without DAPT.

DAPT has shown effectiveness for task-oriented dialog modeling. Wu et al. (2020) further pre-trained BERT on 9 task-oriented dialog corpora and outperformed BERT on four downstream tasks, especially in the few-shot setting. Gu et al. (2020) further pre-trained GPT-2 on 13 dialog corpora ranging from chitchats to task-oriented dialogs, leading to better results on three task-oriented datasets.

However, does further pre-training always help? Mehri et al. (2020) performed DAPT on 700M open-domain dialogs and TAPT, but the resulting model only outperforms BERT in 4 out of 7 task-oriented dialog datasets. We also observe that replacing BERT with TOD-BERT-mlm (Wu et al., 2020) that is further pre-trained MLM on 101K task-oriented dialogs does not always bring a significant difference on downstream tasks. So far, however, there has been little discussion about when and why further pre-training on in-domain data can boost the performance on a downstream task and how the DAPT data size can affect this.

In this paper, we conduct an empirical study on the effect of further pre-training $\text{BERT}_{\text{BASE}}$ on task-oriented dialogs. Our experiments are organized around the following research questions:

- **RQ1** When can DAPT improve the performance on a downstream task?
- **RQ2** How does the amount of data for DAPT affect the performance on a downstream task?

We evaluate further pre-trained models on five downstream tasks involving seven task-oriented dialog datasets. Our main findings are summarized as follows: (1) DAPT and TAPT do not always improve fine-tuning performance: the effect varies for different tasks, models, and fine-tuning data sizes. (2) DAPT is more beneficial in the low-resource setting. As the fine-tuning data size grows, the model's representations change more, implying that the in-

*Corresponding author.
[1]Codes are available at https://github.com/zqwerty/ToDDAPT.

fluence of pre-training decays, thus the benefit of DAPT decreases or even vanishes. (3) Increasing the amount of data for DAPT mostly improves the performance in the relative low-resource setting.

2 Experimental Setup

2.1 Further Pre-training

We further pre-train $\text{BERT}_{\text{BASE}}$ uncased model using masked language modeling loss with 15% tokens masked. Our DAPT dataset consists of several multi-turn task-oriented dialog datasets, including Schema (Rastogi et al., 2020), Taskmaster-1&2 (Byrne et al., 2019), MetaLWOZ (Li et al., 2020), MSR-E2E (Li et al., 2018), SMD (Eric et al., 2017), Frames (El Asri et al., 2017), WOZ (Mrkšić et al., 2017), and Camrest (Wen et al., 2017), which has 103K dialogs (13M words) in total. To investigate **RQ2**, we also use 25%, 5% and 1% dialogs to perform DAPT. For TAPT, we use the training set of each downstream task. We use 95% dialogs for training and select the best checkpoint with the lowest MLM loss on the other 5% dialogs. To obtain a training sample $D_{1:t} = \{U_1, S_1, ..., U_t\}$ where U_i, S_i are user's utterance and system's utterance respectively, we randomly pick a dialog D and sample a turn $t \in [1, T]$ uniformly, where T is the length of D. Then all the utterances are concatenated into a sequence as the model input: "`[CLS]` `[USR]` U_1 `[SEP]` `[SYS]` S_1 `[SEP]` ... `[USR]` U_t `[SEP]`", where `[USR]` and `[SYS]` are two special tokens prepended to user's and system's utterances respectively. See Appendix A for the hyper-parameter setting.

2.2 Evaluation

We conduct comprehensive evaluations on 5 downstream tasks. Models on these tasks are adapted from TOD-BERT (Wu et al., 2020), DialoGLUE (Mehri et al., 2020), or ConvLab-2 (Zhu et al., 2020). See Appendix B for fine-tuning details.

Intent Classification (IC) is a sequence classification problem, where models take an utterance as input and predict its intent. We use three datasets: **HWU** (Liu et al., 2019) that has 64 intents and 26K utterances, **BANKING** (Casanueva et al., 2020) that has 77 intents and 13K utterances, and **OOS** (Larson et al., 2019) that has 151 intents and 24K utterances. We pass the representation of `[CLS]` token to a linear layer for prediction.

Slot Filling (SF) requires models to extract slots' values in an utterance, which is often formulated as a sequence tagging problem. We use **REST8K** dataset (Coope et al., 2020) that has 5 slots and 8K utterances. We add a linear layer on the top of the tokens' representations to predict BIO format tags.

Semantic Parsing (SP) aims at identifying both intents and slots' values in an utterance. We use **TOP** dataset (Gupta et al., 2018) that has 45K utterances spanning 25 intents and 36 slots and **MultiWOZ** 2.3 dataset (Han et al., 2020) that has 10K dialogs and 143K utterances spanning 7 domains, 13 intents, and 25 slots. We use two linear layers to predict intent and tokens' tags respectively.

Dialog State Tracking (DST) is the task of recognizing user constraints throughout the conversation. We use MultiWOZ dataset version 2.1 (Eric et al., 2020) that has 30 domain-slot pairs to track. We adopt two BERT-based models: **TripPy** (Heck et al., 2020) and **TOD-DST** (Wu et al., 2020). Both models use BERT to encode dialog history.

Dialog Act Prediction (DAP) is a multi-label sequence classification problem, where models predict the intents of the system response given the dialog history. We use two datasets: **MultiWOZ** and **GSIM** (Shah et al., 2018) that contains 6 intents and 3K dialogs. For each intent, we feed the representation of `[CLS]` token to a linear layer and predict whether the intent is in the response.

As for evaluation metrics, we use accuracy for intent prediction, macro-F1 for slot filling and dialog act prediction, exact-match for semantic parsing, and joint goal accuracy for dialog state tracking.

2.3 Representational Similarity Analysis

Representational similarity analysis (RSA) is a technique to measure the similarity between models' representations (Laakso and Cottrell, 2000). Following Merchant et al. (2020), we encode samples from a test dataset and randomly select the same $n = 5000$ tokens as stimuli, whose contextual representations at each layer are used to compute an $n \times n$ pairwise cosine similarity matrix. The final similarity score between two models' representations at a certain layer is computed as the Pearson correlation between the flattened upper triangular of the two similarity matrices.

3 Empirical Analysis

3.1 Full Data Experiments

We fine-tune BERT, TOD-BERT-mlm (Wu et al., 2020) that is further pre-trained on 9 task-oriented

	IC			SF	SP		DST (MultiWOZ)		DAP	
	HWU	BANKING	OOS	REST8K	TOP	MultiWOZ	TripPy	TOD-DST	MultiWOZ	GSIM
BERT	91.14	92.61	84.76	95.32	81.49	76.94	58.39	44.63	79.67	45.46
- Std of 3 runs	0.48	0.23	1.21	0.25	0.21	0.14	0.24	0.28	0.44	0.02
TOD-BERT-mlm	91.17	92.82	84.35	95.50	80.86	78.20	58.59	47.66	81.47	45.78
DAPT (all data)	90.80	92.89	84.64	**96.21**	80.96	77.59	58.45	45.71	79.92	45.42
25% data	91.26	91.88	85.55	95.77	80.98	77.71	58.00	**46.32**	**81.80**	45.70
5% data	91.26	**93.08**	84.91	96.03	81.35	77.66	58.06	45.48	80.28	**45.72**
1% data	90.33	91.72	85.64	95.83	**81.81**	**77.93**	58.75	46.08	79.72	45.37
TAPT	**91.91**	92.24	**87.45**	95.76	81.57	77.66	58.49	45.85	80.57	45.56
DAPT+TAPT	91.17	92.89	85.02	96.03	81.17	77.73	**59.12**	45.85	78.92	45.70
$\overline{\Delta}_{\text{DAPT}}$	-0.22	-0.21	0.42	0.63	-0.21	0.79	-0.08	1.27	0.76	0.10
Δ_{TAPT}	0.77	-0.37	2.69	0.43	-0.26	0.72	0.10	1.22	0.90	0.10

Table 1: Performance on downstream tasks. We report the means and standard deviations across three random seeds for BERT. Note that TOD-BERT-mlm has pre-trained on MultiWOZ dataset. A task is in red if further pre-training all outperform BERT by at least one standard deviation. The best task performances are boldfaced.

datasets including MultiWOZ, and our further pre-trained models on downstream tasks using the same hyper-parameters. The results are shown in Table 1. To measure the performance variance, we fine-tune BERT three times using different random seeds and report means and standard deviations. We evaluate the effect of DAPT, denoted by $\overline{\Delta}_{\text{DAPT}}$, through averaging the improvements of the models with different DAPT data sizes (100%, 25%, 5%, 1%) compared with BERT. Similarly, Δ_{TAPT} is the benefit of TAPT.

We find that $\overline{\Delta}_{\text{DAPT}}$ and Δ_{TAPT} sometimes are small or even negative, indicating that **DAPT and TAPT does not always improve performances on downstream tasks.** Consistently, TOD-BERT-mlm does not always outperform BERT significantly. Only on tasks in red in Table 1, the benefits of further pre-training are larger than the standard deviations of BERT. In some cases, interestingly, further pre-training can lead to inferior performance. Even on the same MultiWOZ dataset, further pre-training effects vary according to the model architectures used (TripPy, TOD-DST) and the downstream tasks (SP, DST, DAP). Compared with DAPT and DAPT+TAPT, TAPT obtains similar results but requires much lower training cost, which is worth trying before DAPT.

3.2 RQ1: When can DAPT improve the performance on a downstream task?

Since further pre-training does not help in some cases, we want to explore when DAPT can improve the performance on a downstream task. We first show that DAPT does improve the model's LM ability on downstream datasets (Figure 1), and TAPT can more efficiently improve this ability. This

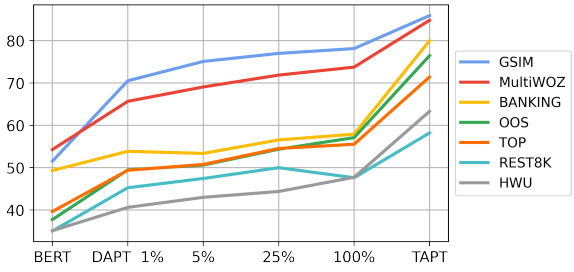

Figure 1: Masked LM prediction accuracy of BERT, DAPT with different data sizes, and TAPT models.

means that further pre-training does reduce the data distribution gap (for LM) between pre-training and fine-tuning but does not guarantee task performance improvement.

A possible hypothesis is that **further pre-training encodes shallow domain knowledge that has obvious influence only when there are insufficient labeled data providing task-specific knowledge for fine-tuning.** By further pre-training, a model learns the co-occurrence of words and their context, which can be viewed as a kind of statistics feature of the target domain. When the fine-tuning data are deficient, this general domain knowledge can alleviate the lack of task-specific knowledge. However, models can learn to encode task-specific knowledge directly through fine-tuning when there are sufficient labeled data and thus rely less on further pre-training.

To verify the hypothesis, we use RSA to assess the representation similarity between fine-tuned models and their initializations for different fine-tuning data sizes. As illustrated in Figure 2, for both BERT and the full data DAPT model, the RSA similarity decreases as the fine-tuning data size grows, especially on the top layers. We ob-

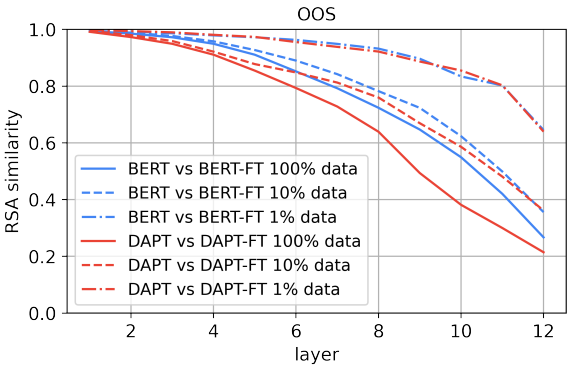

Figure 2: RSA on OOS test set for BERT and DAPT model with 1%, 10%, and 100% fine-tuning data sizes.

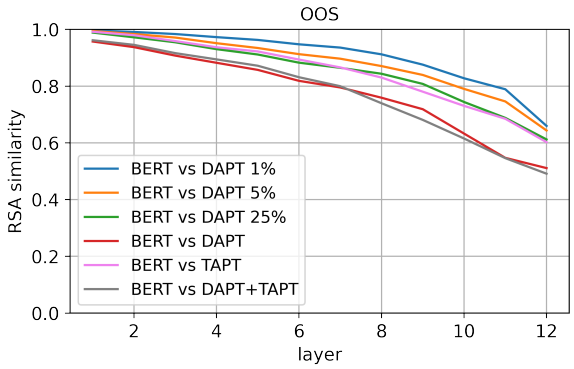

Figure 3: RSA on OOS test set between BERT and further pre-training: 1%, 5%, 25%, and 100% data DAPT, TAPT, and TAPT after full data DAPT.

		$\overline{\Delta}_{\text{DAPT}}$		RSA avg.		best DAPT	
Fine-tune data		100%	10%	100%	10%	100%	10%
IC	HWU	-0.22	0.46	0.73	0.83	5%	5%
	BANKING	-0.21	-0.53	0.74	0.82	5%	25%
	OOS	0.42	1.02	0.69	0.77	1%	25%
SF	REST8K	0.63	0.63	0.73	0.79	100%	25%
SP	TOP	-0.21	0.35	0.71	0.72	1%	5%
	MultiWOZ	0.79	2.44	0.57	0.70	1%	100%
DST	TripPy	-0.08	1.44	0.52	0.56	1%	5%
	TOD-DST	1.27	3.67	0.39	0.49	25%	100%
DAP	MultiWOZ	0.76	0.35	0.26	0.45	25%	100%
	GSIM	0.10	0.28	0.66	0.69	5%	25%

Table 2: Comparison of full and 10% data fine-tuning. We use the same 10% data and average the performance of 3 runs using different random seeds. The RSA similarity averaged across layers is between full data DAPT model and its fine-tuned counterpart. We also report which DAPT data size performs best as "best DAPT".

serve similar trends for all tasks, supporting that less fine-tuning data highlights the importance of pre-training. We also fine-tune the models with and without DAPT in the low-resource setting. Table 2 compares the average performance gain of DAPT ($\overline{\Delta}_{\text{DAPT}}$) and RSA similarity of full data DAPT model and its fine-tuned counterpart (averaged across layers) with full data and low-resource fine-tuning. In the low-resource setting, RSA similarity increases, and DAPT is more beneficial in most cases, which means the knowledge learned through DAPT is more useful when the fine-tuning data are deficient.

3.3 RQ2: How does the amount of data for DAPT affect the performance on a downstream task?

We have shown that models can encode general domain knowledge after DAPT and improve the performance on a downstream task in the low-resource setting. However, how much gain can we obtain by enlarging the DAPT data? To investigate this question, we perform DAPT with 1%, 5%, 25%, and 100% data, ranging from 1020 dialogs to 102K dialogs. From Figure 1, we can see that the more data used in DAPT, the stronger language model on downstream datasets we can get. We also show the change of the model's representation caused by further pre-training in Figure 3. Like fine-tuning, using more data for DAPT brings greater change, and the trend is similar for all tasks. It is also worth noting that compared with DAPT, TAPT changes the model more efficiently in the target dataset.

However, change brought by enlarging DAPT data does not guarantee performance improvement. We compare how much data the best DAPT model used in both full data and low-resource fine-tuning. **As shown in Table 2, including more data for DAPT may not always improve downstream task performance. Nevertheless, when there are less fine-tuning data, the best model needs more data for DAPT.**

4 Conclusion

In this work, we conduct an empirical study to investigate the effect of further pre-training MLM on task-oriented dialogs. Different from earlier findings (Sun et al., 2019; Gururangan et al., 2020), neither DAPT nor TAPT always improves performances on downstream tasks in our experiments. In the low-resource setting, however, DAPT is more helpful, and the size of DAPT data needed to perform best increases. Through RSA, we find that as the fine-tuning data grows, the impact of model initialization fades away, which could be the explanation. We also show that although further pre-

training can improve the model's LM ability on downstream datasets, this may not contribute much to downstream tasks under *pre-train* and *fine-tune* paradigm, calling for novel pre-training objectives and effective ways to use pre-trained models.

Acknowledgments

This work was supported by the National Science Foundation for Distinguished Young Scholars (with No. 62125604) and the NSFC projects (Key project with No. 61936010 and regular project with No. 61876096). This work was also supported by the Guoqiang Institute of Tsinghua University, with Grant No. 2019GQG1 and 2020GQG0005. We would like to thank colleagues from HUAWEI for their constant support and valuable discussion.

References

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4516–4525, Hong Kong, China. Association for Computational Linguistics.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *ArXiv*, abs/2003.04807.

Samuel Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson. 2020. Span-ConveRT: Few-shot span extraction for dialog with pretrained conversational representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 107–121, Online. Association for Computational Linguistics.

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.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219, Saarbrücken, Germany. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *ArXiv*, abs/1907.01669.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 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*, pages 422–428, Marseille, France. European Language Resources Association.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.

Jing Gu, Qingyang Wu, Chongruo Wu, Weiyan Shi, and Zhou Yu. 2020. A tailored pre-training model for task-oriented dialog generation. *ArXiv*, abs/2004.13835.

Abhirut Gupta, Anupama Ray, Gargi Dasgupta, Gautam Singh, Pooja Aggarwal, and Prateeti Mohapatra. 2018. Semantic parsing for technical support questions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3251–3259, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

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 coreference annotation. *ArXiv*, abs/2010.05594.

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.

Aarre Laakso and Garrison Cottrell. 2000. Content and cluster analysis: assessing representational similarity in neural systems. *Philosophical psychology*, 13(1):47–76.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Jinchao Li, Baolin Peng, Sungjin Lee, Jianfeng Gao, Ryuichi Takanobu, Qi Zhu, Minlie Huang, Hannes Schulz, Adam Atkinson, and Mahmoud Adada. 2020. Results of the multi-domain task-completion dialog challenge. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, Eighth Dialog System Technology Challenge Workshop*.

Xiujun Li, Yu Wang, Siqi Sun, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *ArXiv*, abs/1807.11125.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents. In *Proceedings of the Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, Ortigia, Siracusa (SR), Italy. Springer.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *ArXiv*, abs/2009.13570.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 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)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

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

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8689–8696.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *ArXiv*, abs/1801.04871.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 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*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 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*, pages 142–149, Online. Association for Computational Linguistics.

A Pre-Training Details

A.1 Dataset Description

In this section, we present a detailed description of the data we use for further pre-training.

Domain Adaptive Pre-Training (DAPT): We use the pure text of several multi-turn task-oriented dialog datasets for DAPT, including Schema[2] (Rastogi et al., 2020), Taskmaster-1&2[3] (Byrne et al., 2019), MetaLWOZ[4] (Li et al., 2020), MSR-E2E[5] (Li et al., 2018), Frames[6] (El Asri et al., 2017), SMD[7] (Eric et al., 2017), WOZ[8] (Mrkšić et al., 2017), and Camrest[9] (Wen et al., 2017). Table 3 shows the corpus statistics of each dataset. Note that for Schema, we only use its train set as the corpus, and for other datasets, we combine their train, dev, and test sets. For validation, to evaluate the pre-training performance on each corpus separately, we split 5% of the dialogs from each corpus and compute masked language modeling losses on them respectively. For DAPT, we merge the other 95% of each corpus. To reduce the gap between pre-training and fine-tuning, we remove system side utterances at the beginning and the end in each dialog to ensure that the first sentence and the last sentence of each dialog are both from the user side.

Task Adaptive Pre-Training (TAPT): We use the pure text of each downstream task dataset for TAPT and delete the system side utterances at the beginning and the end of each dialog. Similar to DAPT, we use 95% dialogs for training and 5% for validation.

A.2 Hyper-Parameters

In this section, we describe the hyper-parameters we use for further pre-training and how we choose them in our experiments.

	Dialogs	Utterances	Tokens
Schema*	16,142	313,822	3.14M
Taskmaster	30,483	540,311	4.96M
MetalWOZ	40,201	384,381	2.96M
MSR-E2E	10,087	65,451	0.744M
SMD	3,030	13,044	0.116M
Frames	1,369	19,445	0.247M
WOZ	1,200	8,824	1.00M
Camrest	676	4,812	0.0557M
SUM	103,188	1,350,090	13.2M

Table 3: Statistics of pre-training corpus from datasets. The Schema corpus (marked with *) is obtained from the train set and others are obtained by merging train, dev, and test set.

Domain Adaptive Pre-Training (DAPT): In DAPT, we further pre-train the $\text{BERT}_{\text{BASE}}$ uncased model from the official checkpoint in (Devlin et al., 2019) with masked language modeling loss. We use 100%, 25%, 5% and 1% dialogs to perform DAPT respectively. For each setting, we search the hyper-parameters and select the best model according to MLM loss on valid set. We use Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-6$, L2 weight decay of 0.01, and linear decay of the learning rate. We search maximum learning rate in {5e-5, 1e-4, 3e-4}, warmup proportion in {0, 0.06, 0.1}, batch size in {64, 256}, max sequence length in {256, 512}, training steps in {5K, 10K, 20K, 40K}. For other hyper-parameters, we keep them the same as (Devlin et al., 2019). We further pre-train our model on a single Quadro RTX 6000 GPU. It takes 0.3 hours to finish 1K steps pre-training.

Task Adaptive Pre-Training (TAPT): In TAPT, we search the hyper-parameters as in DAPT except that we search training steps in {500, 1K, 2K, 5K, 10K}.

B Fine-Tuning Details

B.1 Dataset Description

In our experiments, we use seven downstream datasets across five tasks, including HWU[10] (Liu et al., 2019), BANKING[11] (Casanueva et al., 2020),

[2] https://github.com/google-research-datasets/dstc8-schema-guided-dialogue

[3] https://github.com/google-research-datasets/Taskmaster

[4] https://www.microsoft.com/en-us/research/project/metalwoz/

[5] https://github.com/xiul-msr/e2e_dialog_challenge

[6] https://www.microsoft.com/en-us/research/project/frames-dataset/#!download

[7] https://nlp.stanford.edu/blog/a-new-multi-turn-multi-domain-task-oriented-dialogue-dataset/

[8] https://github.com/nmrksic/neural-belief-tracker/tree/master/data/woz

[9] https://github.com/zhangzthu/ConvLab2-Pretraining/tree/pretraining/data/camrest

[10] https://github.com/xliuhw/NLU-Evaluation-Data

[11] https://github.com/PolyAI-LDN/task-specific-datasets

Corpus	|Train|	|Dev|	|Test|	Input Format	Labels	Metrics
Intent Classification						
HWU†	8,954	1,076	1,076	Single-Turn	Intent	Accuracy
BANKING†	8,622	1,540	3,080	Single-Turn	Intent	Accuracy
OOS‡	15,100	3,100	5,500	Single-Turn	Intent	Accuracy
Slot Filling						
REST8K†	7,244	1,000	3,731	Single-Turn	Intent, Slot	Macro F1
Semantic Parsing						
TOP†	31,279	4,462	9,042	Single-Turn	Intent, Slot	Exact Match
MultiWOZ 2.3♢	8,434	999	1,000	Multi-Turn	Intent, Slot	Exact Match
Dialog State Tracking						
MultiWOZ 2.1†‡	8,434	999	1,000	Multi-Turn	Dialog State	Joint Goal Accuracy
Dialog Act Prediction						
MultiWOZ 2.1‡	8,434	999	1,000	Multi-Turn	Dialog Act	Macro F1
GSIM‡	1,500	469	1,039	Multi-Turn	Dialog Act	Macro F1

Table 4: Downstream datasets information and the model architecture we used for each dataset. The sizes of train/dev/test sets are the number of dialogs. Note that we mark MultiWOZ 2.1 in dialog state tracking with two symbols, † and ‡, because we adopt two model architectures on this dataset: TripPy (Heck et al., 2020) from DialoGLUE (Mehri et al., 2020) (†) and the DST model from TOD-BERT (Wu et al., 2020) (‡).

OOS[12] (Larson et al., 2019), REST8K[13] (Larson et al., 2019), TOP[14] (Gupta et al., 2018), MultiWOZ 2.1[15] (Eric et al., 2019), MultiWOZ 2.3[16] (Han et al., 2020) and GSIM[17] (Shah et al., 2018). All these datasets are publicly available and can be downloaded directly from the Internet. The datasets information is shown in Table 4.

B.2 Model Architectures for Downstream Tasks

For different downstream tasks, we adopt task-specific model architectures from the three works as listed below and replaced the pre-trained $\text{BERT}_{\text{BASE}}$ with our model. Note that for multi-turn dialog inputs, we reverse the utterances as described in Section 2.1. We keep the original hyper-parameters in each work unchanged when fine-tuning the model.

DialoGLUE DialoGLUE (Mehri et al., 2020) is a benchmark for the language understanding of task-oriented dialogs. Apart from the datasets, DialoGLUE also provides the model architectures built on the pre-trained BERT model for different tasks. The datasets on which we adopt model architectures from DialoGLUE is marked as † in Table 4.

TOD-BERT TOD-BERT (Wu et al., 2020) is a recent model for task-oriented dialogs understanding. We use their models for dialog state tracking and dialog act prediction, marked as ‡ in Table 4.

Convlab-2 Convlab-2 (Zhu et al., 2020) is an open-source toolkit that helps researchers build, evaluate and diagnose task-oriented dialog systems. We mark the dataset on which we use the model architecture from Convlab-2 as ♢ in Table 4.

[12] https://github.com/clinc/oos-eval
[13] https://github.com/PolyAI-LDN/task-specific-datasets
[14] http://fb.me/semanticparsingdialog
[15] https://github.com/budzianowski/multiwoz
[16] https://github.com/budzianowski/multiwoz
[17] https://github.com/google-research-datasets/simulated-dialogue