Efficiently Tuned Parameters are Task Embeddings

Wangchunshu Zhou[1*], **Canwen Xu**[2*], **Julian McAuley**[2]
[1] ETH Zurich [2] University of California, San Diego
[1]wangchunshu.zhou@inf.ethz.ch, [2]{cxu,jmcauley}@ucsd.edu

Abstract

Intermediate-task transfer can benefit a wide range of NLP tasks with properly selected source datasets. However, it is computationally infeasible to experiment with all intermediate transfer combinations, making choosing a useful source task a challenging problem. In this paper, we anticipate that task-specific parameters updated in parameter-efficient tuning methods are likely to encode task-specific information. Therefore, such parameters can be predictive for inter-task transferability. Thus, we propose to exploit these efficiently tuned parameters as off-the-shelf task embeddings for the efficient selection of source datasets for intermediate-task transfer. We experiment with 11 text classification tasks and 11 question answering tasks. Experimental results show that our approach can consistently outperform existing inter-task transferability prediction methods while being conceptually simple and computationally efficient. Our analysis also reveals that the ability of efficiently tuned parameters on transferability prediction is disentangled with their in-task performance. This allows us to use parameters from early checkpoints as task embeddings to further improve efficiency.[1]

1 Introduction

The *pretraining then fine-tuning* paradigm (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2018, 2019; Brown et al., 2020; Lewis et al., 2020; Raffel et al., 2019) has substantially improved the state-of-the-art on a wide range of natural language processing (NLP) tasks. In this paradigm, we first pretrain a large language model on large-scale corpora in a general domain, and then fine-tune the pretrained model to be a task-specific model on the target dataset. In addition to directly transferring from a general pretrained language model, prior work (Phang et al., 2018) also shows that *intermediate-task transfer*, i.e., fine-tuning on intermediate source tasks before the target task, can further improve target task performance. However, the success of intermediate-task transfer heavily relies on the selection of a proper source dataset while an inappropriate source dataset often leads to performance degradation compared to plain fine-tuning. Therefore, some recent works (Vu et al., 2020; Poth et al., 2021) investigate methods to efficiently predict inter-task transferability without actually trying out all intermediate-task combinations.

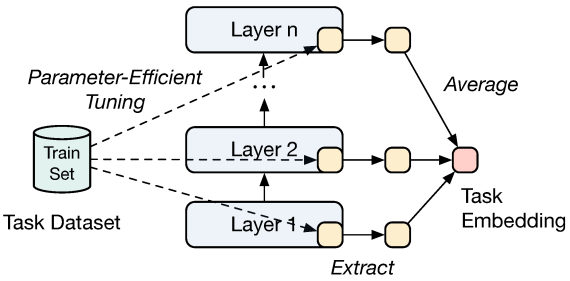

Figure 1: The workflow of using efficiently tuned parameters as task embeddings. The yellow boxes represent tunable parameters in Transformer layers.

The current state of the art (Vu et al., 2020) on predicting inter-task transferability is built on Task2Vec (Achille et al., 2019), which considers the Fisher information matrix of a model fine-tuned on a task as the "task embedding", and predicts inter-task transferability by computing the cosine similarity between the task embedding of the source and target tasks. Despite empirically performing well, this approach requires fine-tuning the full model and (inefficiently) computing the Fisher matrix of the model. Moreover, the resulting task embeddings generally have a high dimensionality similar to the size of the underlying model. Therefore, intermediate task selection, which requires storing task embeddings for each source/target task, can be space-consuming, especially when experi-

[*] Equal contribution.
[1] Code available at https://github.com/JetRunner/TuPaTE.

menting with large language models.

In this work, we opt for parameter-efficient tuning approaches (Houlsby et al., 2019; Li and Liang, 2021; Guo et al., 2021; Hu et al., 2022; Zaken et al., 2022) for the efficient and accurate prediction of inter-task transferability. Our key insight is that task-specific parameters updated in parameter-efficient tuning methods are likely to encode high density task-specific information since they are used as a query for retrieving task-related knowledge in a frozen pretrained language model. Therefore, we propose to directly use task-specific parameters learned via parameter-efficient tuning on source/target datasets as task embeddings, as shown in Figure 1. Compared to task embeddings obtained by calculating the Fisher matrix of the fine-tuned model (Achille et al., 2019; Vu et al., 2020), efficiently tuned parameters are of much lower dimensionality and do not suffer from noise from uninformative weights in the model parameters, thus leading to more accurate transferability prediction. Also, our method only requires parameter-efficient tuning on the tasks and stores task-specific parameters, making both computing and storing task embeddings more efficient. Moreover, with the development of open-source parameter-efficient tuning platforms like AdapterHub (Pfeiffer et al., 2020), we can easily access off-the-shelf parameters of the source and target datasets downloaded from the model zoo and then compute the similarity between the downloaded parameters.

We empirically verify the effectiveness of our approach by experimenting with 11 text classification tasks and 11 question answering tasks, following Vu et al. (2020). Our results show that our approach consistently outperforms existing inter-task transferability prediction methods while being simpler and more efficient. In addition, we find that the ability of efficiently tuned parameters on transferability prediction is not strongly correlated with their in-task performance. Therefore, task-specific parameters tuned with a relatively small number of steps are already highly predictive for inter-task transferability, allowing us to further improve the efficiency of intermediate task selection.

2 Related Work

Prior work (Phang et al., 2018) shows that positive transfer can be elicited by training a model on intermediate source tasks before fine-tuning on the target task. However, the choice of an appropriate source task is crucial for effective transfer. Phang et al. (2018) show that the size of the source dataset is an good prior for source task selection. Pruksachatkun et al. (2020) propose to use task requiring complex reasoning and inference as source tasks. Besides these heuristics, a number of work also focuses on systematic prediction of intermediate task transferability. Vu et al. (2020) propose to used TASK2VEC to construct task embeddings based on the input text or Fisher information matrix of a fine-tuned model. Poth et al. (2021) further extend similar ideas for adapter-based transfer learning. More recently, Vu et al. (2021) explore prompt-based transfer and propose to use prompt similarity as a predictor for prompt transferability to select proper soft prompts for initialization. This can be viewed as a special case of our proposed method where the parameter-efficient tuning method is restricted to vanilla prompt tuning (Lester et al., 2021) and the transfer method is restricted to prompt transfer instead of general intermediate-task transfer.

3 Methodology

3.1 Parameter-Efficient Tuning

Parameter-efficient tuning only updates a small portion of parameters in a large pretrained model. In this paper, we experiment with three types of parameter-efficient tuning: Prompt Tuning (Liu et al., 2021), Bias Tuning (Zaken et al., 2022), and Low-Rank Tuning (Hu et al., 2022).

Prompt Tuning We experiment with P-Tuning v2 (Liu et al., 2021). Specifically, P-Tuning v2 implements a prompt tuning method by introducing additional attention prefix matrices $K_t = \{\mathbf{k}_1 \dots \mathbf{k}_n\}$ and $V_t = \{\mathbf{v}_1 \dots \mathbf{v}_n\}$ for each Transformer layer, where n is a hyperparameter controlling the added prefix length; $\mathbf{k}_*$ and $\mathbf{v}_*$ are vectors with dimension d_h; d_h is the hidden size of the Transformer model.

For each Transformer layer, the added vectors are concatenated with the original key and value matrices to be $K' = K_t \oplus K$ and $V' = V_t \oplus V$, where K and V are the original key and value in each layer's attention block. Then, the new scaled dot-product attention is calculated by replacing the original K and V with the new K' and V', respectively.

Bias Tuning BitFit (Zaken et al., 2022) simply updates all bias terms b in all linear layers $h = Wx + b$ in each Transformer layer.

Low-Rank Tuning LoRA (Hu et al., 2022) injects trainable rank decomposition matrices into each layer of the Transformer model. For each linear layer $h = Wx$ where $W \in \mathbb{R}^{d \times k}$, the forward pass is modified to $h = Wx + BAx$, where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and the rank $r \ll min(d, k)$.

3.2 Tuned Parameters as Task Embeddings

After parameter-efficient tuning, we concatenate all tuned parameters in each Transformer layer and average them across all layers to obtain a vector as a representation for a task, namely **Tu**ned **P**arameters **a**s **T**ask **E**mbedding (TuPaTE). Following Vu et al. (2020), we calculate the cosine similarity between the embeddings of a given targeted task and the candidate source tasks. Then, we rank the candidate source tasks in descending order by the similarity scores.

4 Experiments

4.1 Datasets

Following Vu et al. (2020), we conduct experiments with 11 tasks of text classification or regression (CR) and 11 tasks of question answering (QA). Note that Vu et al. (2020) also includes 11 tasks of sequence labeling. We do not include those datasets since most of them are not publicly available. The list of datasets can be found in Appendix A. To be consistent with Vu et al. (2020), we use two metrics to evaluate the performance of the task embeddings: (1) the average rank ρ of the source task with the *highest* absolute transfer gain; (2) Normalized Discounted Cumulative Gain (NDCG), which is a widely used metric for evaluating the quality of the entire ranking, instead of focusing on the highest rank as ρ does.

4.2 Baselines

We use the following methods as baselines: (1) **DataSize** (Vu et al., 2020) is a simple baseline that ranks all source tasks by the number of training examples. (2) **CurveGrad** (Bingel and Søgaard, 2017; Vu et al., 2020) is a baseline that uses the gradients of the loss curve of BERT for each task. It is originally proposed in Bingel and Søgaard (2017) for predicting gains from multi-task learning and adapted by Vu et al. (2020) for predicting transferability. (3) **TextEmb** (Vu et al., 2020) averages sentence representations over the entire dataset. The sentence representation is obtained by averaging the hidden states in the last layer of BERT. (4) **TaskEmb** (Vu et al., 2020) represents tasks based on the Fisher information matrix. It is adapted from the task embedding originally proposed in Achille et al. (2019) for meta-learning.

Method	#Tuned Param.	Embedding Dim.
TaskEmb	110M	110M
PTuning	184K	15.4K
LoRA	300K	25.0K
BitFit	100K	8.3K

Table 1: Numbers of tuned parameters and the dimensions of the final task representation.

4.3 Training Details

We apply P-Tuning v2, BitFit, and LoRA on BERT-base for fine-tuning on the aforementioned datasets. For each method, we adopt the default hyperparameters from their corresponding papers. Specifically, for P-Tuning v2, we use a prefix length of 20 and search the learning rate from {1e-2, 1e-3}; For LoRA, we set LoRA's r to 8 and α to 8, and search a learning rate from {5e-4, 2e-4}; For BitFit, we search a learning rate from {1e-4, 4e-4}. We train all models with a batch size of 32 for 20 epochs on all datasets. We use the parameters tuned for 2 epochs as "early" task embeddings and those corresponding to the best validation set performance as "late" task embeddings. We compare the number of tunable parameters and the final task embedding dimensions in Table 1. We can see that TuPaTE has a significantly lower dimensionality compare to the TaskEmb baseline. We also include an ensemble of the three efficient tuning methods (denoted as "3 Ensemble"), by averaging the inter-task similarity scores of each model.

4.4 Experimental Results

We present the main results in Table 2. We find that TuPaTE with different parameter-efficient tuning methods consistently outperforms prior works including TextEmb and TaskEmb. Interestingly, the performance improvement is larger in Full → Limited and Limited → Limited settings. We conjecture that this is because in limited resource

Task Type	Method	FULL → FULL				FULL → LIMITED				LIMITED → LIMITED			
		in-class (10)		*all-class (21)*		*in-class (10)*		*all-class (21)*		*in-class (10)*		*all-class (21)*	
		$\rho\downarrow$	NDCG↑	$\rho\downarrow$	NDCG↑	$\rho\downarrow$	NDCG↑	$\rho\downarrow$	NDCG↑	$\rho\downarrow$	NDCG↑	$\rho\downarrow$	NDCG↑
Classification/ Regression (CR)	DATASIZE	3.6	80.4	7.8	75.2	3.8	62.9	8.9	57.2	-	-	-	-
	CURVEGRAD	5.5	68.6	-	-	6.4	45.2	-	-	5.9	50.8	-	-
	TEXTEMB	5.2	76.4	9.8	74.7	3.5	60.3	7.5	55.6	4.8	61.4	11.4	46.2
	TASKEMB	2.8	82.3	5.4	78.3	3.4	68.2	7.1	63.5	4.2	62.6	9.7	47.7
	TUPATE												
	+PTUNING	2.5	83.7	4.5	81.0	**3.1**	71.3	6.4	65.1	3.9	64.6	8.1	51.3
	+LORA	2.7	83.0	5.0	79.6	3.3	70.5	6.8	63.7	4.0	64.2	9.0	49.3
	+BITFIT	2.5	83.5	4.3	81.6	3.2	71.1	6.5	64.6	**3.8**	64.9	8.3	50.9
	3 ENSEMBLE	**2.3**	**83.9**	**4.2**	**81.8**	**3.1**	**71.5**	**6.2**	**65.3**	**3.8**	**65.1**	**8.0**	**51.5**
Question Answering (QA)	DATASIZE	3.2	84.4	11.4	65.8	2.3	77.0	11.2	43.5	-	-	-	-
	CURVEGRAD	8.3	64.8	-	-	8.2	49.1	-	-	6.8	53.4	-	-
	TEXTEMB	4.1	81.1	5.8	82.0	2.7	77.6	3.8	80.5	4.1	65.6	7.3	69.1
	TASKEMB	3.2	84.5	5.4	82.8	2.5	78.0	3.6	81.6	3.6	67.1	7.1	69.5
	TUPATE												
	+PTUNING	3.0	85.7	**4.8**	83.3	2.2	80.9	3.1	83.5	**3.2**	**68.3**	**6.3**	**72.4**
	+LORA	3.1	85.3	5.2	83.0	2.3	79.8	3.3	82.5	3.4	67.5	6.7	70.8
	+BITFIT	3.0	85.5	4.9	83.1	2.1	81.4	3.1	83.4	3.3	68.0	6.5	72.0
	3 ENSEMBLE	**2.9**	**85.9**	**4.8**	**83.5**	**2.0**	**81.7**	**2.9**	**83.7**	**3.2**	68.2	**6.3**	**72.4**

Table 2: To evaluate TuPaTE, we measure the average rank (ρ) assigned to the best source task (i.e., the one that results in the largest transfer gain) across target tasks, as well as the average NDCG measure of the overall ranking's quality. Parentheses denote the number of source tasks in each setting. Some results of CURVEGRAD are missing (marked with "-") since its code is not available. The other results of CURVEGRAD are taken from Vu et al. (2020).

Method	$\Delta\rho$	ΔNDCG	NCDG-Perf. Pearson
PTUNING	0.0	+0.1	0.25
LORA	0.0	−0.1	0.17
BITFIT	+0.1	+0.2	0.20

Table 3: Analysis on the correlation between task-specific performance (e.g., accuracy) and transferability prediction results (i.e., ρ and NDCG) for different parameter-efficient tuning methods. $\Delta\rho$ and ΔNDCG denote the difference of ρ and NDCG between the parameters with the highest and lowest task-specific performance.

Method	Early		Best	
	ρ	NDCG	ρ	NDCG
PTUNING	2.5	83.5	2.5	83.7
LORA	2.8	82.6	2.7	83.0
BITFIT	2.5	83.2	2.5	83.5

Table 4: Transferability prediction results with early checkpoints (checkpoints after 2 epochs) and the best checkpoints (checkpoints corresponding to the best validation performance).

settings, parameter-efficient tuning methods generally perform much better than full fine-tuning, which is used in the TASKEMB method. Moreover, we find that PTUNING and BITFIT outperform LORA in all settings. We suspect this is because the amount of tunable parameters in LORA is much larger than PTUNING and BITFIT. Also, the ensemble of three methods achieve even better performance than only using one approach.

4.5 Analysis

We conduct additional experiments in the *in-class* setting on classification/regression tasks to better understand how TuPaTE works. We first analyze the correlation between the in-task performance (e.g., accuracy) and transferability prediction ability of efficiently tuned parameters. We train TuPaTE with 5 random combinations between searchable hyperparameters and random seeds, and present the correlation in Table 3. We observe that there is only a weak correlation between in-task performance and transferability prediction results, indicating that the ability of efficiently tuned parameters to encode task-related information is disentangled with their final in-task performance. This also shows the robustness of TuPaTE with respect to hyperparameters.

The fact that in-task performance only correlates weakly with transferability prediction motivates us to explore whether early checkpoints of

efficiently tuned parameters can be used for transferability prediction. From Table 4, we find that early checkpoints are also effective task embeddings. This allows us to reduce the computation cost by around 90% while substantially outperforming the TASKEMB baseline.

5 Conclusion

In this paper, we show that efficiently tuned parameters are highly predictive for inter-task transferability and thus can be used as off-the-shelf task embeddings for source task selection in intermediate-task transfer learning. Our empirical investigation with three parameter-efficient tuning methods on 22 NLP tasks demonstrates that our approach outperforms prior works on inter-task transferability prediction despite being more efficient.

Limitations

We select three representative works for three types of parameter-efficient tuning. However, there are other parameter-efficient tuning methods that we have not investigated. Although we believe our conclusion can generalize to other methods, we will conduct more experiments to confirm for future work.

Ethics Statement

We propose to use efficiently tuned parameters as task embedding, only for predicting the performance of intermediate transfer learning. Thus, we do not anticipate any major ethical concern.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments. This project is partly supported by NSF Award #1750063.

References

Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. 2019. ComQA: A community-sourced dataset for complex factoid question answering with paraphrase clusters. 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 307–317. Association for Computational Linguistics.

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2vec: Task embedding for meta-learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6429–6438. IEEE.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2503–2514. The COLING 2016 Organizing Committee.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *MLCW*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

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. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378. Association for Computational Linguistics.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First Quora Dataset Release: Question pairs.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5189–5197. AAAI Press.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Hector J. Levesque. 2011. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*. AAAI.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*, pages 7871–7880. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? efficient intermediate task selection. *arXiv preprint arXiv:2104.08247*.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R Bowman. 2020. Intermediate-task transfer learning

with pretrained models for natural language understanding: When and why does it work? *arXiv preprint arXiv:2005.00628*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200. Association for Computational Linguistics.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2021. Spot: Better frozen model adaptation through soft prompt transfer. *arXiv preprint arXiv:2110.07904*.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380. Association for Computational Linguistics.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL*, pages 1–9. Association for Computational Linguistics.

A List of Datasets

Task	|Train|
Text classification / Regression (CR)	
SNLI (Bowman et al., 2015)	570k
MNLI (Williams et al., 2018)	393k
QQP (Iyer et al., 2017)	364k
QNLI (Wang et al., 2019)	105k
SST-2 (Socher et al., 2013)	67k
SciTail (Khot et al., 2018)	27k
CoLA (Warstadt et al., 2019)	8.5k
STS-B (Cer et al., 2017)	7k
MRPC (Dolan and Brockett, 2005)	3.7k
RTE (Dagan et al., 2005)	2.5k
WNLI (Levesque, 2011)	634
Question Answering (QA)	
SQuAD-2 (Rajpurkar et al., 2018)	162k
NewsQA (Trischler et al., 2017)	120k
HotpotQA (Yang et al., 2018)	113k
SQuAD-1 (Rajpurkar et al., 2016)	108k
DuoRC-p (Saha et al., 2018)	100k
DuoRC-s (Saha et al., 2018)	86k
DROP (Dua et al., 2019)	77k
WikiHop (Welbl et al., 2018)	51k
BoolQ (Clark et al., 2019)	16k
ComQA (Abujabal et al., 2019)	11k
CQ (Bao et al., 2016)	2k

Table 5: The datasets used in our experiments and their training set size (Vu et al., 2020).