The Power of Prompt Tuning for Low-Resource Semantic Parsing

Nathan Schucher[1,2] **Siva Reddy**[2,3] **Harm de Vries**[1]
[1]ElementAI, a ServiceNow company
[2]Mila/McGill University
[3]Facebook CIFAR AI Chair
{nathan.schucher,harm.devries}@servicenow.com

Abstract

Prompt tuning has recently emerged as an effective method for adapting pre-trained language models to a number of language tasks. In this paper, we investigate prompt tuning for semantic parsing, the task of mapping natural language utterances onto formal meaning representations. For large T5 models we find (i) that prompt tuning significantly outperforms fine-tuning in the low data regime and (ii) that canonicalization—i.e. naturalizing the meaning representations—barely improves performance. This last result is surprising as it suggests that large T5 models can be modulated to generate sequences that are far from the pre-training distribution.

1 Introduction

With the widespread success of large language models (LMs; Devlin et al. 2018; Raffel et al. 2020; Bommasani et al. 2021), it becomes increasingly important to explore how such models can be adapted to downstream tasks. One emerging adaptation capability of LMs like GPT-3 is that their behaviour can be modulated by a text-prompt (Brown et al., 2020). That is, a wide variety of novel tasks can be successfully performed by prompting the model with a task description and a few examples. Nevertheless, this adaptation strategy is sensitive to the exact phrasing of the prompt, and, more importantly, performs worse than fine-tuning models on task-specific examples (Lester et al., 2021).

While it has been explored to automatically search for discrete prompts (Shin et al., 2020), tuning "soft" or continuous prompts has recently arisen as strong performing alternative (Lester et al., 2021). Prompt tuning only optimizes the embeddings of a number of prompt tokens while keeping all other LM parameters frozen. On a set of language understanding tasks, Lester et al. (2021) showed that this method becomes competitive with fine-tuning for the largest pre-trained T5 models (Raffel et al., 2020). Li and Liang (2021) also explore a related parameter-efficient adaptation method called prefix-tuning, finding that it outperforms fine-tuning on a number of language generation tasks.

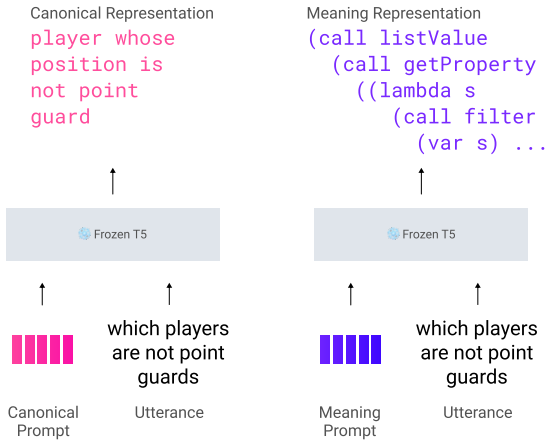

Figure 1: We show that prompt tuning large T5 models achieve almost similar performance on canonical (left) and meaning representations (right), even when only 200 Overnight examples are provided.

In this paper, we investigate prompt tuning for semantic parsing. This task is fundamentally different from aforementioned language understanding and generation tasks, as it requires to output formal meaning representations, which do not resemble the natural language distribution seen during pre-training. We especially focus on the low-resource setup because examples for semantic parsing are difficult and expensive to collect (Wang et al., 2015; Marzoev et al., 2020). We evaluate prompt tuning on two datasets: the 200-shot version of Overnight (Wang et al., 2015; Shin et al., 2021) and the low-resource splits TOPv2 (Chen et al., 2020). On both datasets, we compare prompt tuning T5 against fine-tuning and investigate the effect of canonicalizing the meaning representation, i.e. to what extent naturalizing the logical forms influences performance. In addition, we study the

effect of T5 model scale on Overnight while we investigate different data regimes on TOPv2. Our main findings can be summarized as follows:

- For large T5 models, prompt tuning significantly outperforms fine-tuning in the low-data regime. This performance gap decreases when more training data becomes available.
- In contrast to previous work (Shin et al., 2021), we find that canonicalizing the meaning representations has little effect on the performance, even in low-data regimes.

This last result suggests that T5-large can be modulated to output sequences that are much further from the pre-training distribution than previously thought, and presents additional evidence for the view that large pre-trained transformers are universal computation engines (Lu et al., 2021).

2 Background

Our work is related to recent work on semantic parsing and prompt tuning, which we briefly describe below.

2.1 Semantic Parsing

Semantic parsing is the task of converting a natural language utterance $\mathbf{u} = (u_1, \ldots, u_N)$ to a formal meaning representation $\mathbf{z} = (z_1, \ldots, z_M)$. These meaning representations, also referred to as logical forms, can be interpreted by machines and executed in a real environment. For example, ThingTalk (Campagna et al., 2019) and TOP (Gupta et al., 2018) are meaning representations for executing commands of virtual assistants, while SQL is a representation for interacting with relational databases.

In recent years, neural sequence-to-sequence models have become the dominant approach for semantic parsing tasks (Dong and Lapata, 2016). Such models learn to encode a natural language utterance $\mathbf{u}$ into a sequence of continuous embeddings, which are then decoded into a probability distribution over the meaning representation $\mathbf{z}$. The parameters of these encoder-decoder models are trained to maximize the likelihood of outputting the correct representation on the training set. To ensure the decoder only outputs valid meaning representations, some works have explored methods for constraining the output space of the decoder (Cheng et al., 2017; Yin and Neubig, 2018; Lin et al., 2019).

Large language models Recent work has explored how to leverage pre-trained language models, like BERT (Devlin et al., 2018), BART (Lewis et al., 2020), and T5 (Raffel et al., 2020), for semantic parsing tasks. As these language models are trained on text-only corpora, it is unclear to what extent they can be adapted to generate meaning representations. Some researchers have only used BERT in the encoder (Wang et al., 2020; Scholak et al., 2020; Xu et al., 2020a). Others have proposed to use a generative model like BART to augment the dataset by paraphrasing natural language utterances (Xu et al., 2020b). Recently, it has been shown that T5 can be successfully fine-tuned on a large-scale text-to-sql dataset (Shaw et al., 2021; Scholak et al., 2021).

Canonicalization A common simplification step in semantic parsing is to canonicalize the meaning representations. That is, the meaning representation $\mathbf{z}$ is naturalized to a canonical form $\mathbf{c}$ through a grammar or set of rules. An example of the meaning and canonical representation for Overnight (Wang et al., 2015) can be found in Fig. 1.

When canonical representations are available, Berant and Liang (2014) argue that semantic parsing can be seen as a paraphrase task. They propose to use a paraphrase model—using e.g. word vectors trained on Wikipedia—to find the best paraphrase of utterance $\mathbf{u}$ among a set of canonical utterances. They show this paraphrase model improves results over directly generating logical forms on two question-answering datasets. Marzoev et al. (2020) extends this work by showing that pre-trained language models like BERT can be effective paraphrasers. While Berant and Liang (2014); Marzoev et al. (2020) use models to score canonical utterances, Shin et al. (2021) propose to constrain the generation process of autoregressive models like BART and GPT-3. On a number of few-shot semantic parsing tasks, they demonstrate the benefit of generating canonical representations over meaning representations.

2.2 Prompt Tuning

This paper is inspired by the recent work of Lester et al. (2021) and Li and Liang (2021), which shows that a simple adaptation technique called prompt tuning is very effective in modulating a pre-trained language model. Prompt tuning prepends a sequence of tokens $\mathbf{p} = (p_1, \ldots, p_K)$ to the input

sequence $\mathbf{u} = (u_1, \ldots, u_N)$ before it is fed to a language model (with parameters θ). Prompt tokens p_k are added to the vocabulary and each has a continuous embedding $e(p_k)$. During prompt tuning we optimize the embeddings $(e(p_1), \ldots, e(p_K))$ while keeping the language model parameters θ fixed. Note that this process still requires to backpropagate the gradients through the full language model. Like fine-tuning other sequence-to-sequence models, we maximize the likelihood of generating the output sequence $\mathbf{z}$.

Lester et al. (2021) evaluates prompt tuning on SuperGLUE, a benchmark consisting of eight language understanding tasks. They find that prompt tuning becomes competitive with fine-tuning for the largest T5 model. Li and Liang (2021) propose prefix-tuning to adapt BART and GPT-2 for natural language generation tasks. This method differs from Lester et al. (2021) in that it prepends trainable embeddings for each layer of the language model rather than introducing token embeddings at the input layer. They demonstrate that pre-fix outperforms fine-tuning baselines. Similarly, Liu et al. (2021) also show encouraging results for prompt tuning on natural language understand and generation tasks. Qin and Eisner (2021) also explores prompt tuning but for a knowledge extraction task. Inserting general adapter layers into pre-trained language models is also proposed in Houlsby et al. (2019); Mahabadi et al. (2021). Related to our work are also other few-shot adaptation techniques like PET (Schick and Schütze, 2020; Schick and Schütze, 2020). Moreover, adapter layers have also been explored in the computer vision domain (Rebuffi et al., 2017; De Vries et al., 2017).

3 Experiments

To evaluate low-resource prompt tuning, we compare against fine-tuned variants of the same model on two semantic parsing datasets with canonical representations available: Overnight and TOPv2. We compare both large and small variants of the T5 architecture on these datasets and experiment with various canonicalized representations.

3.1 Datasets

We first provide more details on the Overnight (Wang et al., 2015) and the TOPv2 (Chen et al., 2020) datasets.

Overnight The Overnight semantic parsing dataset consists of 13,682 natural utterance, canonical form, meaning representation triples split across eight domains. The task is to map a natural language utterance to the corresponding meaning representation. The dataset was collected by first enumerating all pairs of canonical utterances and meaning representations from a synchronous context-free grammar to a fixed depth. These human-interpretable, but unnatural utterances, were subsequently paraphrased by crowd-sourced workers into natural language utterances.

To simulate low-resource splits of this dataset, we follow Shin et al. and create randomly subsampled splits of 200 training examples for each domain, using 20% of the remaining data for validation. We measure and report denotation accuracy by evaluating all predicted queries using the SEMPRE toolkit (Berant et al., 2013). We repeat each experiment on Overnight with five different random splits.

TOPv2 Chen et al. (2020) introduce the TOPv2 dataset, a task-oriented semantic parsing dataset with eight domains, two of which come with predefined low-resource splits. The authors propose a principled way of constructing low-resource training sets, *samples per intent and slot* (SPIS), intended to ensure equal exposure to ontology labels across domains of varying complexity. We experiment with the *weather* and *reminder* domains at the 10, 25, and 500 SPI splits, performing five runs on each resource-level varying the random seed. The *reminder* domain is the most challenging with 19 intent labels, 32 slot labels, and with 21% of the programs having a depth greater than 2. *Weather* in comparison has 7 intent labels, 11 slot labels, and no programs with depth greater than 2.

3.2 Canonicalized Representations

Next, we review and compare the canonicalized representations used in the Overnight and TOPv2 datasets.

3.2.1 Overnight

Overnight uses a context-free synchronous grammar to generate canonical representations for the logical forms. As can be seen in Figure 1, these canonical representations of overnight resemble natural language. While the original paper used these synthetic utterances as templates for crowd-sourcing natural language utterances, subsequent work has used them as targets for semantic parsing models (Marzoev et al., 2020; Shin et al., 2021).

We follow this work and treat these paraphrased utterances as our target canonicalized representations to compare with the normal Overnight meaning representations.

3.2.2 TOPv2

On the other hand, Chen et al. apply a set of simple modifications to the TOPv2 meaning representations to arrive at a canonical form used in all their experiments. Unlike Overnight, these preprocessing steps are largely small encoding differences and do not change the syntactic structure of the logical forms. We adopt all of these canonicalization steps except for lexicographic sorting of the semantic parse tree with an additional ontology label shortening step. Examples of these transformations can be seen in Figure 2 and are briefly described below.

Utterance removes redundant utterance tokens unnecessary for interpreting the meaning representation. In Figure 2 the tokens "Driving directions to" are captured by the attached `IN:GET_DIRECTIONS` intent label, whereas the tokens attached to the leaf slots (`Eagles`, `game`) are semantically relevant and must be kept.

Vocab adds the intent and slot labels to the language model tokenizer and embedding lookup table as atomic tokens. These will appear as randomly initialized emedding vectors and are not trained during prompt tuning.

Token replaces the intent and slot labels with a short unique identifier (e.g. `T1`) tokenizable by the pre-trained tokenizer and embeddings.

We perform an ablation over these choices repeating each experiment three times varying the random seed.

3.3 Models

We experiment with BART and T5, two large pretrained encoder-decoder language models (Lewis et al., 2020; Raffel et al., 2020). BART is trained on the same 160GB text dataset used to train RoBERTa (Lewis et al., 2020) with a denoising objective. There are two size configurations (BART-base, BART-large) and we experiment only with the 406M parameter BART-large on the Overnight dataset. T5 is trained on the 750GB C4 dataset (Raffel et al., 2020) with a de-noising objective.

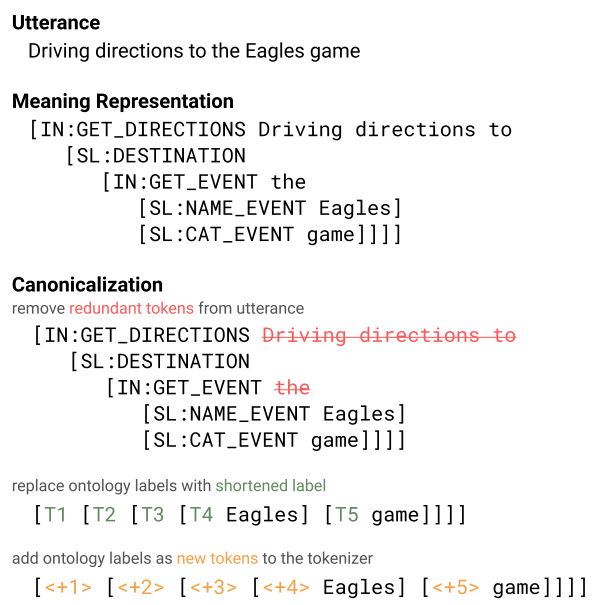

Figure 2: Example from TOPv2 dataset with different canonicalization strategies applied.

We use the T5-v1.1 checkpoints from Lester et al. (2021) that were trained for an additional 100K steps with the Prefix-LM objective. T5-v1.1 has five configurations at various scales: small, base, large, xl, xxl which have 60M, 220M, 770M, 3B, and 11B parameters, respectively. Here, we experiment with models up to T5-large.

Fine-tuning baseline We compare against baselines that fine-tune all parameters of BART and T5. We train the T5 models with AdaFactor (Shazeer and Stern, 2018) and BART with Adam (Lewis et al., 2020). On TOPv2, we use a learning rate of 1e-4 and batch size of 128. On Overnight, we use a learning rate of 1e-3 and a batch size of 64 across all sizes of T5. On both datasets, we train for 5000 epochs and perform model selection by early stopping on the validation set.

Prompt tuning We follow the prompt tuning procedure proposed by Lester et al. for T5. We use 150 prompt tokens for all model sizes with a learning rate of 0.3 optimized with AdaFactor. We train for 5000 epochs on most domains, although as high as 20000 on the low-resource splits. Like the fine-tuned baseline, we perform model selection with best exact match accuracy on the validation set. We apply the same method to BART and found that it did not converge under a number of hyperparameter configurations. We therefore exclude prompt

Model	Representation	Method	Basketball	Blocks	Calendar	Housing	Publications	Recipes	Restaurants	Social	Average
T5-small	Canonical	FT	0.775	0.466	0.721	0.616	0.665	0.673	0.636	0.568	0.640
		PT	0.756	0.466	0.698	0.598	0.658	0.680	0.668	0.562	0.636
	Meaning	FT	0.767	0.454	0.685	0.608	0.640	0.698	0.691	0.581	0.641
		PT	0.601	0.338	0.519	0.446	0.470	0.477	0.626	0.392	0.484
T5-base	Canonical	FT	0.800	0.466	0.736	0.642	0.711	0.694	0.696	0.597	0.668
		PT	0.792	0.484	0.724	0.652	0.699	0.689	0.713	0.597	0.669
	Meaning	FT	0.771	0.455	0.717	0.612	0.670	0.713	0.714	0.587	0.655
		PT	0.629	0.453	0.683	0.590	0.656	0.673	0.743	0.471	0.612
BART	Canonical	FT	0.591	0.331	0.740	0.309	0.668	0.598	0.582	0.532	0.544
	Meaning	FT	0.734	0.370	0.514	0.540	0.514	0.477	0.417	0.424	0.499
T5-large	Canonical	FT	0.793	0.458	0.760	0.658	0.678	0.727	0.715	0.581	0.671
		PT	0.816	0.526	**0.800**	**0.661**	0.733	0.758	**0.803**	**0.663**	**0.720**
	Meaning	FT	0.777	0.432	0.690	0.639	0.709	0.729	0.723	0.590	0.661
		PT	0.818	0.531	0.786	0.654	**0.735**	0.752	0.792	0.657	0.716
GPT-3†	Canonical	Context	0.760	0.460	0.680	0.560	0.580	0.740	0.740	0.550	0.634
	Meaning	Context	0.560	0.390	0.500	0.420	0.460	0.660	0.580	0.480	0.506
BART†	Canonical	FT	**0.852**	**0.539**	0.726	0.656	0.714	0.773	0.756	0.585	0.700
	Meaning	FT	0.813	0.476	0.732	0.566	0.696	**0.778**	0.720	0.536	0.665

Table 1: Denotation accuracy for all models on the Overnight dataset. For each domain, we report the average over 5 runs with different randomly sampled splits of 200 examples. † are reported from Shin et al. (2021)

SPIS	model	method	reminder	weather	average
10	T5-large	FT	0.392	0.579	0.486
		PT	**0.567**	**0.700**	**0.634**
25	BART-CopyPtr	FT	0.557	0.716	0.637
	T5-large	FT	0.502	0.683	0.593
		PT	**0.642**	**0.739**	**0.691**
500	BART-CopyPtr	FT	0.719	**0.849**	0.784
	T5-large	FT	0.649	0.846	0.748
		PT	**0.749**	0.847	**0.798**

Table 2: Average exact match accuracies (5 runs) for different low-resource splits of the TOPv2 dataset.

tuned BART models from our results[1].

4 Results

We report all Overnight results in Table 1. We display the results of T5-large on the three different SPI splits of TOPv2 in Table 2. In the Method column of both tables FT and PT indicate fine-tuned or prompt tuned versions of each model, respectively. In Table 3, we summarize the results of the canonicalization ablation study for TOPv2.

4.1 Prompt tuning vs fine-tuning

We find that prompt tuning improves over fine-tuning for all large model configurations and target representations. Averaged over all overnight domains, we see up to 7% improvement in denotation accuracy with T5-large. For T5-small and T5-base, prompt tuning remains competitive with fine-tuning, and is within 1% average accuracy when predicting canonical forms. On TOPv2, we observe that prompt tuning achieves an absolute improvement of 15% over fine-tuning on the lowest SPIS split (again averaged over both domains). This gap shrinks to 5% as the dataset size increases to the 500 SPIS.

Our prompt tuning models outperform previously reported results on these datasets. On Overnight, our best model—T5-large PT with canonical representations—outperforms the BART FT model of Shin et al. (2021) by 2 accuracy points. On the 25 SPIS split of TOPv2, we see an average improvement of more than 5 points compared to the BART-CopyPTR of Chen et al. (2020).

Prompt tuning reveals a difference in T5 model capabilities at different scales that may be obscured by fine-tuning. For fine-tuned T5-models, the absolute difference between meaning and canonical representations is relatively consistent as the model size grows. For prompt tuned models, however, we see the accuracy of meaning representations rapidly increase from a 15% deficit (T5-small) to within 1% (T5-large).

4.2 Canonical vs meaning representations

For T5-large, we find very small gains for generating canonical representations instead of meaning representations. On overnight (Table 1), the gap is 0.4 accuracy points for T5-large. This difference is much smaller than previously reported for unconstrained decoding with BART and GPT-3 by Shin et al. (2021) (3 and 13 points, respectively). For our BART fine-tuning baseline we also observe a gap of more than 4 points between canonical and meaning representations. Interestingly, we do not find such a gap for canonical and meaning representations when fine-tuning T5 models, indicating there are qualitative differences between BART and

[1] Li and Liang also find that prompt tuning with BART is unstable and parameterize the prefix with an MLP; we did not attempt this setup.

SPIS	Model	None	Tokens	Tokens + Utterance	Utterance	Vocab	Vocab + Utterance
10	T5-small	0.434 ± 0.163	0.445 ± 0.102	0.301 ± 0.069	0.309 ± 0.06	0.234 ± 0.265	0.196 ± 0.239
	T5-large	**0.7** ± **0.02**	0.641 ± 0.077	0.673 ± 0.035	0.663 ± 0.025	0.689 ± 0.013	0.585 ± 0.116
25	T5-small	0.555 ± 0.135	0.549 ± 0.112	0.512 ± 0.07	0.51 ± 0.085	0.27 ± 0.297	0.276 ± 0.315
	T5-large	0.739 ± 0.013	0.713 ± 0.04	**0.746** ± **0.012**	0.726 ± 0.031	0.699 ± 0.047	0.721 ± 0.014
500	T5-small	0.718 ± 0.118	0.722 ± 0.122	0.727 ± 0.122	0.723 ± 0.121	0.483 ± 0.344	0.423 ± 0.307
	T5-large	0.847 ± 0.004	0.852 ± 0.007	**0.859** ± **0.003**	0.855 ± 0.01	0.833 ± 0.002	0.81 ± 0.003

Table 3: Exact match accuracies (3 runs) on TOPv2 dataset for different meaning representation canonicalization choices. **bold** indicates best exact match accuracy at that resource level, underline indicates best representation per model.

T5 models.

On TOPv2, we find a similar pattern that large T5 models can extrapolate far beyond the training distribution. We especially observe this when adding novel vocabulary tokens (`Vocab` in Table 3), arguably the most out-of-distribution canonicalization choice. These novel token embeddings are randomly initialized and never updated during prompt tuning. Nevertheless, a prompt tuned T5-large model can successfully generate such sequences whereas a prompt tuned T5-small model struggles. Specifically, T5-large only sees a modest decrease from 70% to 68.9% on 10 SPIs and from 84.7.9% to 83.3% on the 500 SPIS (see `None` and `Vocab` columns in Table 3, respectively). In contrast, T5-small drops from 43.4% to 23.4% on 10 SPIs and from 71.8% to 48.3% on 500 SPIs.

Interestingly, we find that `Token` drastically reduces performance for T5-small at the 10 SPIS level—30.9% vs. 43.4% for `None`—but slightly outperforms it at 500 SPIS. We speculate that `Token`, effectively anonymizes the ontology tokens, obscuring information that can be useful for prediction. In low-data regimes there is not enough training data to learn the semantics of these anonymized tokens, whereas with enough data this problem vanishes and the shorter target sequence ultimately is easier to predict.

5 Conclusion

We find that prompt tuning is an effective method for adapting language models to the semantic parsing task. Prompt tuning significantly outperforms fine-tuning in low-data regimes, and remains competitive in fully-supervised settings. This is in spite of the fact that the semantic parsing tasks considered often require generating sequences far from the pre-training distribution. We find that canonicalizing meaning representations does not significantly improve performance and that targeting the normal meaning representation is sufficient when fine-tuning or prompt tuning large T5 models.

We find evidence of a phase-transition in the ability of language models to generate sequences far from their training distribution. This scaling property is obscured by the fine-tuning process—which easily adapts the model parameters to the downstream task. Prompt tuning modulates the language model in such a way that this property is not destroyed. We speculate that there are likely other interesting qualitative differences in large models that are worth investigating and that prompt tuning is a promising tool for doing so.

References

J. Berant, A. Chou, R. Frostig, and P. Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland. Association for Computational Linguistics.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. 2021. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

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. *arXiv:2005.14165 [cs]*.

Giovanni Campagna, Silei Xu, Mehrad Moradshahi, Richard Socher, and Monica S. Lam. 2019. Genie: A generator of natural language semantic parsers for virtual assistant commands. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2019, page 394–410, New York, NY, USA. Association for Computing Machinery.

Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. 2020. Low-Resource Domain Adaptation for Compositional Task-Oriented Semantic Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5090–5100, Online. Association for Computational Linguistics.

Jianpeng Cheng, Siva Reddy, Vijay Saraswat, and Mirella Lapata. 2017. Learning structured natural language representations for semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 44–55, Vancouver, Canada. Association for Computational Linguistics.

Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron Courville. 2017. Modulating early visual processing by language. *arXiv preprint arXiv:1707.00683*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Cite arxiv:1810.04805Comment: 13 pages.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, Brussels, Belgium. 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 *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. *arXiv:2104.08691 [cs]*.

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, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. *arXiv:2101.00190 [cs]*.

Kevin Lin, Ben Bogin, Mark Neumann, Jonathan Berant, and Matt Gardner. 2019. Grammar-based neural text-to-sql generation. *CoRR*, abs/1905.13326.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. 2021. Pretrained transformers as universal computation engines. *arXiv preprint arXiv:2103.05247*.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. *arXiv preprint arXiv:2106.04647*.

Alana Marzoev, Samuel Madden, M. Frans Kaashoek, Michael J. Cafarella, and Jacob Andreas. 2020. Unnatural language processing: Bridging the gap between synthetic and natural language data. *CoRR*, abs/2004.13645.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*.

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.

S-A Rebuffi, H. Bilen, and A. Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*.

Timo Schick and Hinrich Schütze. 2020. Few-shot text generation with pattern-exploiting training. *arXiv preprint arXiv:2012.11926*.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few-shot text classification and natural language inference. *Computing Research Repository*, arXiv:2001.07676.

Torsten Scholak, Raymond Li, Dzmitry Bahdanau, Harm de Vries, and Chris Pal. 2020. DuoRAT: Towards Simpler Text-to-SQL Models. *arXiv:2010.11119 [cs]*.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. Picard: Parsing incrementally for constrained auto-regressive decoding from language models.

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? 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 922–938, Online. Association for Computational Linguistics.

Noam M. Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *ArXiv*, abs/1804.04235.

Richard Shin, Christopher H. Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained Language Models Yield Few-Shot Semantic Parsers. *arXiv:2104.08768 [cs]*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online. Association for Computational Linguistics.

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a Semantic Parser Overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1332–1342, Beijing, China. Association for Computational Linguistics.

Silei Xu, Giovanni Campagna, Jian Li, and Monica S. Lam. 2020a. Schema2qa: High-quality and low-cost q&a agents for the structured web. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 1685–1694. ACM.

Silei Xu, Sina Semnani, Giovanni Campagna, and Monica Lam. 2020b. AutoQA: From databases to QA semantic parsers with only synthetic training data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 422–434, Online. Association for Computational Linguistics.

Pengcheng Yin and Graham Neubig. 2018. TRANX: A transition-based neural abstract syntax parser for semantic parsing and code generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 7–12, Brussels, Belgium. Association for Computational Linguistics.