# The Power of Prompt Tuning for Low-Resource Semantic Parsing

**Nathan Schucher**[1,2] **Siva Reddy**[2,3] **Harm de Vries**[1]
[1]ServiceNow Research
[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 understanding and generation tasks. In this paper, we investigate prompt tuning for semantic parsing—the task of mapping natural language utterances onto formal meaning representations. On the low-resource splits of Overnight and TOPv2, we find that a prompt tuned T5-xl significantly outperforms its fine-tuned counterpart, as well as strong GPT-3 and BART baselines. We also conduct ablation studies across different model scales and target representations, finding that, with increasing model scale, prompt tuned T5 models improve at generating target representations that are far from the pre-training distribution.

## 1 Introduction

With the widespread success of pre-trained language models (LMs; Devlin et al. 2019; Raffel et al. 2020; Bommasani et al. 2021), it becomes increasingly important to explore how such models can be adapted to downstream tasks. One adaptation method which has recently attracted much attention is prompt design (Brown et al., 2020; Shin et al., 2020), which modulates the behaviour of a LM through a task description and a few input-output examples. Brown et al. (2020) show that this adaptation strategy is increasingly effective for larger LMs. However, prompt design 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).

Prompt tuning has recently arisen as a strong performing alternative adaption method (Lester et al., 2021). Rather than hand-designing discrete prompts, prompt tuning optimizes the embeddings of a number of task-specific prompt tokens. In contrast to fine-tuning, this method keeps almost all LM parameters frozen. On a set of language understanding tasks, Lester et al. (2021) show that prompt tuning 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 low-resource natural language generation tasks.

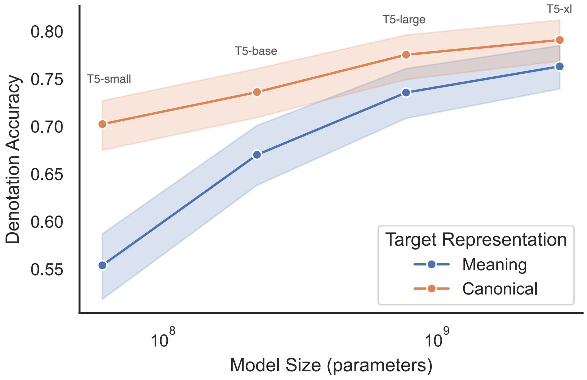

Figure 1: We show that the T5 prompt tuning performance difference between target representations shrinks as the number of parameters increase, with constrained decoded T5-xl achieving close to performance parity.

In this paper, we investigate prompt tuning for semantic parsing. This task is fundamentally different from the aforementioned language understanding and generation tasks, as it requires that models output formal meaning representations which do not resemble the natural language distribution seen during pre-training. In particular, we 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 therefore 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

*Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*
*Volume 2: Short Papers*, pages 148 - 156
May 22-27, 2022 ©2022 Association for Computational Linguistics

representation, i.e. to what extent naturalizing the logical forms influences performance. In addition, we study the effect of T5 model scale on Overnight as well as varying 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, resulting in an absolute improvement of 6% and 15% on Overnight and TOPv2, respectively. This performance gap decreases when more training data becomes available.

- With growing model size, prompt tuned T5 models are increasingly capable of outputting diverse target representations (see Figure 1). On Overnight, we find that the disparity between canonical and meaning representations shrinks from 17% to 4% for T5-small and T5-xl, respectively. On TOPv2, prompt tuned T5-large models are much better at generating out-of-vocabulary tokens than T5-small.

## 2 Related work

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).

**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. Examples of the meaning and canonical representation for Overnight and TOPv2 (Wang et al., 2015; Chen et al., 2020) can be found in Fig. 2.

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

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, 2021). 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. We compare both large

TOPv2

Natural Language Utterance

Driving directions to the Eagles game

Meaning Representation

```
[IN:GET_DIRECTIONS Driving directions to
   [SL:DESTINATION
      [IN:GET_EVENT the
         [SL:NAME_EVENT Eagles]
         [SL:CAT_EVENT game]]]]
```

Canonicalized Representation

simplify meaning representation by removing utterance tokens

[IN:GET_DIRECTIONS ~~Driving directions to~~<br>
[SL:DESTINATION<br>
[IN:GET_EVENT ~~the~~<br>
[SL:NAME_EVENT Eagles]<br>
[SL:CAT_EVENT game]]]]

replace ontology labels with short label of existing In-Vocab tokens

```
[T1 [T2 [T3 [T4 Eagles] [T5 game]]]]
```

add ontology labels to tokenizer as single Out-of-Vocab token

```
[<+1> [<+2> [<+3> [<+4> Eagles] [<+5> game]]]]
```

Overnight

Natural Language Utterance

which players are not point guards

Meaning Representation

```
(call listValue
  (call getProperty
    ((lambda s
       (call filter
         (var s)
         (string position)
         (string !=) en.position.point_guard))
      (call domain
        (string player)))
    (string player)))
```

Canonicalized Representation

corresponding grammar-generated canonicalized form

```
player whose position is not point guard
```

Figure 2: Examples from the TOPv2 and Overnight datasets with the corresponding canonicalization schemes.

and small variants of the T5 architecture on these datasets and experiment with various canonicalized representations.

### 3.1 Datasets

**Overnight** The Overnight semantic parsing dataset (Wang et al., 2015) consists of 13,682 natural utterance, canonical form, meaning representation triples split across eight domains. 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 SPIS resource splits, performing five runs on each model 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

#### 3.2.1 Overnight

Overnight uses a context-free synchronous grammar to generate canonical representations for the logical forms. As can be seen in Fig. 2, these canonical representations resemble natural language.

#### 3.2.2 TOPv2

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 pre-processing 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) and add an ontology label shortening step. Examples of these transformations can be seen in Fig. 2 and are briefly described below.

**Simplify** removes redundant utterance tokens unnecessary for interpreting the meaning representation.

**Out-of-Vocab** adds the entire intent or slot label to the tokenizer as a new single tokens with a corresponding randomly initialized embedding.

**In-Vocab** replaces the intent and slot labels with a short unique identifier representable by the pre-trained tokenizer.

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

| Model | Representation | Method | Bask. | Blo. | Cal. | Hou. | Pub. | Rec. | Res. | Soc. | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| T5-small | Meaning | FT | 0.767 | 0.454 | 0.685 | 0.608 | 0.640 | 0.698 | 0.691 | 0.581 | 0.641 |
| | | PT | 0.621 | 0.312 | 0.470 | 0.352 | 0.478 | 0.506 | 0.608 | 0.352 | 0.463 |
| | Canonical | FT | 0.775 | 0.466 | 0.721 | 0.616 | 0.665 | 0.673 | 0.636 | 0.568 | 0.640 |
| | | PT | 0.764 | 0.440 | 0.680 | 0.601 | 0.648 | 0.699 | 0.697 | 0.578 | 0.638 |
| T5-base | Meaning | FT | 0.769 | 0.455 | 0.717 | 0.612 | 0.670 | 0.713 | 0.714 | 0.587 | 0.655 |
| | | PT | 0.717 | 0.429 | 0.677 | 0.510 | 0.596 | 0.639 | 0.705 | 0.492 | 0.596 |
| | Canonical | FT | 0.800 | 0.466 | 0.736 | 0.642 | 0.711 | 0.694 | 0.696 | 0.597 | 0.668 |
| | | PT | 0.786 | 0.452 | 0.682 | 0.636 | 0.675 | 0.705 | 0.733 | 0.614 | 0.660 |
| BART | Meaning | FT | 0.734 | 0.370 | 0.514 | 0.540 | 0.514 | 0.477 | 0.417 | 0.424 | 0.499 |
| | Canonical | FT | 0.591 | 0.331 | 0.740 | 0.309 | 0.668 | 0.598 | 0.582 | 0.532 | 0.544 |
| T5-large | Meaning | FT | 0.777 | 0.432 | 0.690 | 0.639 | 0.709 | 0.729 | 0.723 | 0.590 | 0.661 |
| | | PT | 0.792 | 0.469 | 0.739 | 0.676 | 0.696 | 0.734 | 0.778 | 0.600 | 0.685 |
| | Canonical | FT | 0.793 | 0.458 | 0.760 | 0.658 | 0.678 | 0.727 | 0.715 | 0.581 | 0.671 |
| | | PT | 0.819 | 0.525 | 0.768 | 0.712 | 0.744 | 0.789 | 0.769 | 0.655 | 0.723 |
| T5-xl | Meaning | FT | 0.774 | 0.413 | 0.702 | 0.630 | 0.682 | 0.691 | 0.705 | 0.580 | 0.647 |
| | | PT | 0.819 | 0.532 | 0.767 | 0.693 | 0.694 | 0.758 | 0.778 | 0.632 | 0.709 |
| | Canonical | FT | 0.799 | 0.486 | **0.781** | 0.647 | 0.724 | 0.732 | 0.725 | 0.619 | 0.689 |
| | | PT | **0.839** | **0.544** | 0.777 | **0.729** | **0.770** | **0.791** | **0.789** | **0.702** | **0.743** |

Table 1: Unconstrained denotation accuracy for all models (with unconstrained decoding) on the Overnight dataset. For each domain, we report the average over 5 runs trained on randomly sampled splits of 200 examples for fine-tuned (FT) and prompt tuned (PT) models.

### 3.3 Models

We provide training details and hyperparameters for all models in Appendix A. Below, we briefly explain the prompt-tuning methodology.

#### 3.3.1 Prompt Tuning

Prompt tuning, as proposed by Lester et al. (2021), prepends a sequence of continuous embeddings $\mathbf{p} = (p_1, \ldots, p_K)$ to the sequence input embeddings $e(\mathbf{u}) = (e(u_1), \ldots, e(u_N))$ before feeding it to a language model with parameters $\theta$. During prompt tuning we optimize the prompt embeddings $(p_1, \ldots, p_K)$ exclusively, keeping the language model parameters $\theta$ and the pretrained vocabulary embeddings fixed. Note that this process still requires backpropagating gradients through the full language model. Like fine-tuning models, we maximize the likelihood of generating the output sequence $\mathbf{z}$.

## 4 Results

In Table 1, we report Overnight results across four T5 model scales and two target representations. In Table 2, we add constrained decoding (see Appendix A) to our best performing T5 model and compare against previously reported Overnight results. In Table 3, we display the results of T5-large on the three different SPIS-splits of TOPv2, and include the BART-CopyPtr results from Chen et al. (2020). In Table 4, 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. On Overnight, prompt tuned denotation accuracy exceeds fine-tuned counterparts by up to 5 points with T5-large and T5-xl. For T5-small and T5-base, prompt tuning remains competitive (within 1% average accuracy) with fine-tuning when predicting canonical forms. On TOPv2, prompt tuning achieves an absolute improvement of 15% mean accuracy over fine-tuning on the lowest SPIS split. This performance disparity lessens when training data increases; however, prompt tuned T5-large continues to beat its fine-tuned counterpart by 5 points at 500 SPIS and the BART-CopyPtr model by 1.4 points.

Our prompt tuning models outperform previously reported results on these datasets. On Overnight, our best model—T5-xl PT with canonical representations and constrained decoding—outperforms the BART FT model of Shin et al. (2021) by 5 accuracy points, and GPT-3 by more than 2 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).

### 4.2 Canonical vs meaning representations

Our main finding is that prompt tuned T5 models become better at generating meaning representations with increased model size. On Overnight, we see the absolute difference between canonical and meaning representations shrink from 17.5 points

| Model | Representation | Method | Decoding | Bask. | Blo. | Cal. | Hou. | Pub. | Rec. | Res. | Soc. | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T5-xl | Meaning | PT | Constrained | 0.841 | 0.592 | 0.802 | 0.765 | 0.776 | 0.814 | 0.789 | 0.725 | 0.763 |
| T5-xl | Canonical | PT | Constrained | 0.856 | 0.619 | 0.806 | **0.779** | **0.824** | 0.830 | 0.822 | **0.793** | **0.791** |
| BART[†] | Meaning | FT | Constrained | 0.834 | 0.499 | 0.750 | 0.619 | 0.739 | 0.796 | 0.774 | 0.620 | 0.704 |
| BART[†] | Canonical | FT | Constrained | **0.864** | 0.554 | 0.780 | 0.672 | 0.758 | 0.801 | 0.801 | 0.666 | 0.737 |
| GPT-2[†] | Meaning | FT | Constrained | 0.760 | 0.479 | 0.736 | 0.571 | 0.645 | 0.699 | 0.660 | 0.606 | 0.644 |
| GPT-2[†] | Canonical | FT | Constrained | 0.836 | 0.540 | 0.766 | 0.666 | 0.715 | 0.764 | 0.768 | 0.623 | 0.710 |
| GPT-3[†] | Canonical | Context | Constrained | 0.859 | **0.634** | 0.792 | 0.741 | 0.776 | 0.792 | 0.840 | 0.687 | 0.765 |
| GPT-3[†*] | Meaning | Context | Constrained | 0.680 | 0.530 | 0.680 | 0.580 | 0.630 | 0.750 | 0.780 | 0.630 | 0.657 |
| GPT-3[†*] | Canonical | Context | Constrained | 0.800 | 0.620 | **0.820** | 0.710 | 0.790 | **0.840** | **0.890** | 0.720 | 0.774 |

Table 2: Constrained denotation accuracy for all models on the Overnight dataset. For each domain, we report the average over 5 runs trained on randomly sampled splits of 200 examples. [†] denotes results reported by Shin et al. (2021). [*] indicates performance on subsampled test set.

| 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 3: Average exact match accuracies (5 runs) for different low-resource splits of the TOPv2 dataset. BART-CopyPtr results from Chen et al. (2020).

| | None | | Simplified | | In-Vocab | | Out-of-Vocab | |
|---|---|---|---|---|---|---|---|---|
| SPIS | Sm. | Lg. | Sm. | Lg. | Sm. | Lg. | Sm. | Lg. |
| 10 | 0.43 | **0.70** | 0.31 | 0.66 | 0.45 | 0.64 | 0.23 | 0.69 |
| 25 | 0.56 | **0.74** | 0.51 | 0.73 | 0.55 | 0.71 | 0.27 | 0.70 |
| 500 | 0.72 | **0.85** | 0.72 | **0.85** | 0.72 | **0.85** | 0.48 | 0.83 |

Table 4: Exact match accuracies (3 runs) on TOPv2 Weather domain for different meaning representation canonicalization choices (**bold** indicates best exact match accuracy at that resource level), Sm. and Lg. refer to T5-small and T5-large, respectively.

for T5-small to 3.4 points for T5-xl (Table 1). This gap shrinks another 18% to 2.8 points when we apply constrained decoding to T5-xl (Table 2). By contrast, Shin et al. (2021) reports an 11.7 point difference when prompting GPT-3. For our fine-tuning baselines, we observe a small performance gap of 4 points across target representations for BART and T5-xl, while we observe no gap for T5-small, T5-base, and T5-large models.

In our TOPv2 experiments we find similar evidence of large T5 model flexibility for generating sequences far from the training distribution. In particular, for our most intrusive canonicalization scheme `Out-of-Vocab`, which adds novel tokens to the vocabulary and leaves these embeddings un-trained, we find no significant reduction in performance for T5-large across all data resource levels. T5-small, in comparison, sees almost a 50% drop in performance relative to no canonicalization (`None`) at the 10 SPIS level and continues to underperform by 33 % at the 500 SPIS level.

Interestingly, we find that `In-Vocab` 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 `In-Vocab` effectively anonymizes the ontology tokens, obscuring information that is 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.

## 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 the fully supervised setting. We furthermore find that while canonicalizing meaning representations can slightly improve performance, the disparity between target representations decreases when prompt tuning larger T5 models. This result differs from previous work (Shin et al., 2021) which suggested that pre-trained LMs are much better equipped to output canonical than meaning representations. However, a significant limitation of prompt tuning is that it takes more time to converge than fine-tuning. We believe one fruitful direction for future research is to find ways to reduce the compute required to prompt tune.

## 6 Ethical Considerations and Limitations

There are two main limitations of this work. The first is the limited analysis of the learned prompts. While concurrent work has shown that interpreting prompts is a difficult task, it is still an important consideration and left for future work (Khashabi et al., 2021). Secondly, training prompts on meaning representations requires substantially more compute than fine-tuning. This may exacerbate inequalities in regions where access to data and compute are similarly limited (Ahia et al., 2021).

## References

Orevaoghene Ahia, Julia Kreutzer, and Sara Hooker. 2021. The low-resource double bind: An empirical study of pruning for low-resource machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3316–3333.

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.

Harm de Vries, Florian Strub, Jeremie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. 2017. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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.

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.

Daniel Khashabi, Shane Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sameer Singh, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, and Yejin Choi. 2021. PROMPT WAYWARDNESS: The Curious Case of Discretized Interpretation of Continuous Prompts. *arXiv:2112.08348 [cs]*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. 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, Online. Association for Computational Linguistics.

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

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

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. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4596–4604. PMLR.

Richard Shin, Christopher 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. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7699–7715, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

## A Models

Here we provide all model details and hyperparameters to reproduce our results. We experiment with BART and T5 (Lewis et al., 2020; Raffel et al., 2020), two large pre-trained encoder-decoder language models. 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. 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-xl. All experiments were run with PyTorch (v. 1.8.1) and the Huggingface Transformers (v. 4.8.2) library (Paszke et al., 2019; Wolf et al., 2020).

**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; Kingma and Ba, 2015). On TOPv2, we use a learning rate of $10^{-4}$ and batch size of 128. On Overnight, we use a learning rate of $10^{-3}$ 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 it sometimes took as many as 20000 epochs to converge 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 tuned BART models from our results[1].

**Constrained Decoding** We implement grammar-constrained decoding by building a prefix tree containing all canonical or meaning representations in the dataset as in Shin et al. (2021). When doing constrained decoding we perform a beam search with 10 beams and use the prefix tree to look up valid single token continuations of the decoded sequence.

## B Results

For completeness, we provide all Overnight results in Table 5.

### B.1 Training Times

Prompt tuned parameter efficiency comes at a cost: we find that prompt tuning takes significantly longer to train with early stopping than does fine-tuning. On the Overnight dataset, fine-tuned models typically took 250 epochs before validation performance plateaued. Our prompt tuned models frequently took more than 1000 epochs when predicting canonical representations, and up to 5,000 when predicting meaning representations. In Figure 3, we show example training curves for prompt tuning and fine-tuning.

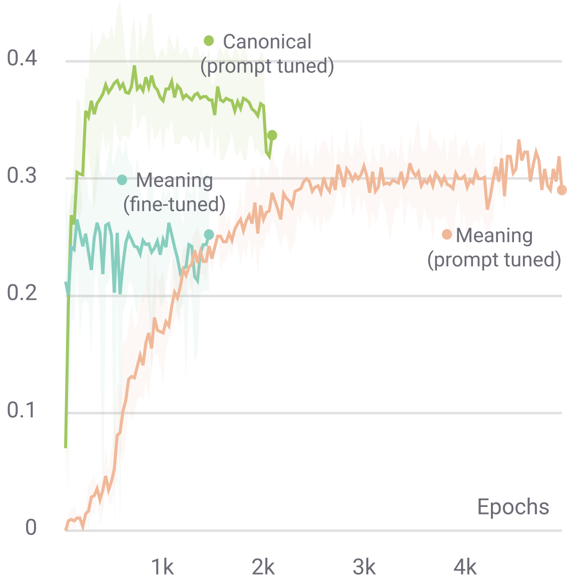

Figure 3: Prompt and fine-tuned exact match validation accuracy on the Overnight *blocks* domain. Fine-tuned models can quickly reach peak validation accuracy regardless of target representation. Prompt tuned models can take thousands of epochs to converge when predicting meaning representations.

[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.

| Model | Representation | Method | Decoding | Bask. | Blo. | Cal. | Hou. | Pub. | Rec. | Res. | Soc. | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T5-small | Meaning | FT | Unconstrained | 0.767 | 0.454 | 0.685 | 0.608 | 0.640 | 0.698 | 0.691 | 0.581 | 0.641 |
| | | | Constrained | 0.787 | 0.519 | 0.725 | 0.624 | 0.753 | 0.752 | 0.705 | 0.664 | 0.691 |
| | | PT | Unconstrained | 0.621 | 0.312 | 0.470 | 0.352 | 0.478 | 0.506 | 0.608 | 0.352 | 0.463 |
| | | | Constrained | 0.656 | 0.392 | 0.615 | 0.475 | 0.593 | 0.588 | 0.663 | 0.450 | 0.554 |
| | Canonical | FT | Unconstrained | 0.775 | 0.466 | 0.721 | 0.616 | 0.665 | 0.673 | 0.636 | 0.568 | 0.640 |
| | | | Constrained | 0.811 | 0.519 | 0.744 | 0.663 | 0.729 | 0.723 | 0.692 | 0.671 | 0.694 |
| | | PT | Unconstrained | 0.764 | 0.440 | 0.680 | 0.601 | 0.648 | 0.699 | 0.697 | 0.578 | 0.638 |
| | | | Constrained | 0.787 | 0.521 | 0.730 | 0.679 | 0.735 | 0.748 | 0.746 | 0.674 | 0.703 |
| T5-base | Meaning | FT | Unconstrained | 0.769 | 0.455 | 0.717 | 0.612 | 0.670 | 0.713 | 0.714 | 0.587 | 0.655 |
| | | | Constrained | 0.790 | 0.496 | 0.738 | 0.639 | 0.743 | 0.745 | 0.737 | 0.644 | 0.692 |
| | | PT | Unconstrained | 0.717 | 0.429 | 0.677 | 0.510 | 0.596 | 0.639 | 0.705 | 0.492 | 0.596 |
| | | | Constrained | 0.754 | 0.494 | 0.760 | 0.593 | 0.725 | 0.699 | 0.752 | 0.586 | 0.670 |
| | Canonical | FT | Unconstrained | 0.800 | 0.466 | 0.736 | 0.642 | 0.711 | 0.694 | 0.696 | 0.597 | 0.668 |
| | | | Constrained | 0.840 | 0.525 | 0.745 | 0.676 | 0.773 | 0.736 | 0.734 | 0.696 | 0.716 |
| | | PT | Unconstrained | 0.786 | 0.452 | 0.682 | 0.636 | 0.675 | 0.705 | 0.733 | 0.614 | 0.660 |
| | | | Constrained | 0.826 | 0.550 | 0.774 | 0.717 | 0.780 | 0.764 | 0.770 | 0.708 | 0.736 |
| BART | Meaning | FT | Unconstrained | 0.734 | 0.370 | 0.514 | 0.540 | 0.514 | 0.477 | 0.417 | 0.424 | 0.499 |
| | Canonical | FT | Unconstrained | 0.591 | 0.331 | 0.740 | 0.309 | 0.668 | 0.598 | 0.582 | 0.532 | 0.544 |
| T5-large | Meaning | FT | Unconstrained | 0.777 | 0.432 | 0.690 | 0.639 | 0.709 | 0.729 | 0.723 | 0.590 | 0.661 |
| | | | Constrained | 0.789 | 0.475 | 0.713 | 0.662 | 0.743 | 0.754 | 0.717 | 0.641 | 0.687 |
| | | PT | Unconstrained | 0.792 | 0.469 | 0.739 | 0.676 | 0.696 | 0.734 | 0.778 | 0.600 | 0.685 |
| | | | Constrained | 0.816 | 0.533 | 0.774 | 0.742 | 0.760 | 0.787 | 0.793 | 0.680 | 0.736 |
| | Canonical | FT | Unconstrained | 0.793 | 0.458 | 0.760 | 0.658 | 0.678 | 0.727 | 0.715 | 0.581 | 0.671 |
| | | | Constrained | 0.819 | 0.509 | 0.751 | 0.703 | 0.718 | 0.742 | 0.728 | 0.664 | 0.704 |
| | | PT | Unconstrained | 0.819 | 0.525 | 0.768 | 0.712 | 0.744 | 0.789 | 0.769 | 0.655 | 0.723 |
| | | | Constrained | 0.841 | 0.597 | 0.805 | 0.770 | 0.794 | 0.823 | 0.823 | 0.750 | 0.775 |
| T5-xl | Meaning | FT | Unconstrained | 0.774 | 0.413 | 0.702 | 0.630 | 0.682 | 0.691 | 0.705 | 0.580 | 0.647 |
| | | | Constrained | 0.799 | 0.453 | 0.731 | 0.658 | 0.749 | 0.724 | 0.728 | 0.647 | 0.686 |
| | | PT | Unconstrained | 0.819 | 0.532 | 0.767 | 0.693 | 0.694 | 0.758 | 0.778 | 0.632 | 0.709 |
| | | | Constrained | 0.841 | 0.592 | 0.802 | 0.765 | 0.776 | 0.814 | 0.789 | 0.725 | 0.763 |
| | Canonical | FT | Unconstrained | 0.799 | 0.486 | 0.781 | 0.647 | 0.724 | 0.732 | 0.725 | 0.619 | 0.689 |
| | | | Constrained | 0.818 | 0.555 | 0.783 | 0.705 | 0.763 | 0.770 | 0.752 | 0.703 | 0.731 |
| | | PT | Unconstrained | 0.839 | 0.544 | 0.777 | 0.729 | 0.770 | 0.791 | 0.789 | 0.702 | 0.743 |
| | | | Constrained | 0.856 | 0.619 | 0.806 | **0.779** | **0.824** | 0.830 | 0.822 | **0.793** | **0.791** |
| BART | Meaning | FT | Unconstrained | 0.813 | 0.476 | 0.732 | 0.566 | 0.696 | 0.778 | 0.720 | 0.536 | 0.665 |
| | | | Constrained | 0.834 | 0.499 | 0.750 | 0.619 | 0.739 | 0.796 | 0.774 | 0.620 | 0.704 |
| | Canonical | FT | Unconstrained | 0.852 | 0.539 | 0.726 | 0.656 | 0.714 | 0.773 | 0.756 | 0.585 | 0.700 |
| | | | Constrained | **0.864** | 0.554 | 0.780 | 0.672 | 0.758 | 0.801 | 0.801 | 0.666 | 0.737 |
| GPT-2 | Meaning | FT | Constrained | 0.760 | 0.479 | 0.736 | 0.571 | 0.645 | 0.699 | 0.660 | 0.606 | 0.644 |
| | Canonical | FT | Constrained | 0.836 | 0.540 | 0.766 | 0.666 | 0.715 | 0.764 | 0.768 | 0.623 | 0.710 |
| GPT-3 | Canonical | Context | Constrained | 0.859 | **0.634** | 0.792 | 0.741 | 0.776 | 0.792 | 0.840 | 0.687 | 0.765 |
| GPT-3* | Meaning | Context | Unconstrained | 0.560 | 0.390 | 0.500 | 0.420 | 0.460 | 0.660 | 0.580 | 0.480 | 0.506 |
| | | | Constrained | 0.680 | 0.530 | 0.680 | 0.580 | 0.630 | 0.750 | 0.780 | 0.630 | 0.657 |
| | Canonical | Context | Unconstrained | 0.760 | 0.460 | 0.680 | 0.560 | 0.580 | 0.740 | 0.740 | 0.550 | 0.634 |
| | | | Constrained | 0.800 | 0.620 | **0.820** | 0.710 | 0.790 | **0.840** | **0.890** | 0.720 | 0.774 |

Table 5: Results across all model size, target representation, tuning method, and decoding method for Overnight dataset. BART, GPT-2, and GPT-3 results results are included from Shin et al. (2021)