SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer

Tu Vu[1,2]★ **Brian Lester**[1] **Noah Constant**[1] **Rami Al-Rfou**[1] **Daniel Cer**[1]
Google Research[1]
University of Massachusetts Amherst[2]
{ttvu,brianlester,nconstant,rmyeid,cer}@google.com
tuvu@cs.umass.edu

Abstract

There has been growing interest in parameter-efficient methods to apply pre-trained language models to downstream tasks. Building on the PROMPTTUNING approach of Lester et al. (2021), which learns task-specific soft prompts to condition a frozen pre-trained model to perform different tasks, we propose a novel prompt-based transfer learning approach called SPOT: **S**oft **P**rompt **T**ransfer. SPOT first learns a prompt on one or more source tasks and then uses it to initialize the prompt for a target task. We show that SPOT significantly boosts the performance of PROMPTTUNING across many tasks. More remarkably, across all model sizes, SPOT matches or outperforms standard MODELTUNING (which fine-tunes all model parameters) on the SUPERGLUE benchmark, while using up to 27,000× fewer task-specific parameters. To understand where SPOT is most effective, we conduct a large-scale study on task transferability with 26 NLP tasks in 160 combinations, and demonstrate that many tasks can benefit each other via prompt transfer. Finally, we propose an efficient retrieval approach that interprets task prompts as task embeddings to identify similar tasks and predict the most transferable source tasks for a novel target task.

1 Introduction

The past few years have seen the rapid development of ever larger pre-trained language models, where it has repeatedly been shown that scaling up the model size is a key ingredient for achieving the best performance (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020). While this trend has continued to push the boundaries of possibility across various NLP benchmarks, the sheer size of these models presents a challenge for their practical application. For 100B+ parameter models, fine-tuning and deploying a separate instance of the model for each downstream task would be prohibitively expensive. To get around the infeasibility of fine-tuning, Brown et al. (2020) propose PROMPTDESIGN, where every downstream task is cast as a language modeling task and the *frozen* pre-trained model performs different tasks by conditioning on manual text prompts provided at inference time. They demonstrate impressive few-shot performance with a single frozen GPT-3 model, although its performance depends highly on the choice of the prompt (Zhao et al., 2021) and still lags far behind state-of-the-art fine-tuning results.

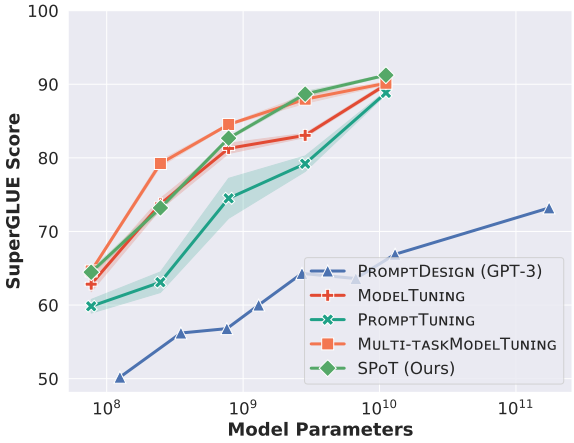

Figure 1: Our SPOT approach—which transfers a prompt learned from a mixture of source tasks (here, GLUE) onto target tasks—outperforms vanilla PROMPTTUNING (Lester et al., 2021) and GPT-3 (Brown et al., 2020) on SUPERGLUE by a large margin, matching or outperforming MODELTUNING across all model sizes. At the XXL model size, SPOT even outperforms MULTI-TASKMODELTUNING, which fine-tunes the entire model on the GLUE mixture before fine-tuning it on individual SUPERGLUE tasks. See Appendix A for full results.

More recent work explores methods for learning *soft prompts* (Liu et al., 2021b; Qin and Eisner, 2021; Li and Liang, 2021; Lester et al., 2021), which can be seen as additional learnable parameters injected into the language model. Lester et al. (2021) propose PROMPTTUNING, a simple method

★ Work done during an internship at Google Research.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5039 - 5059

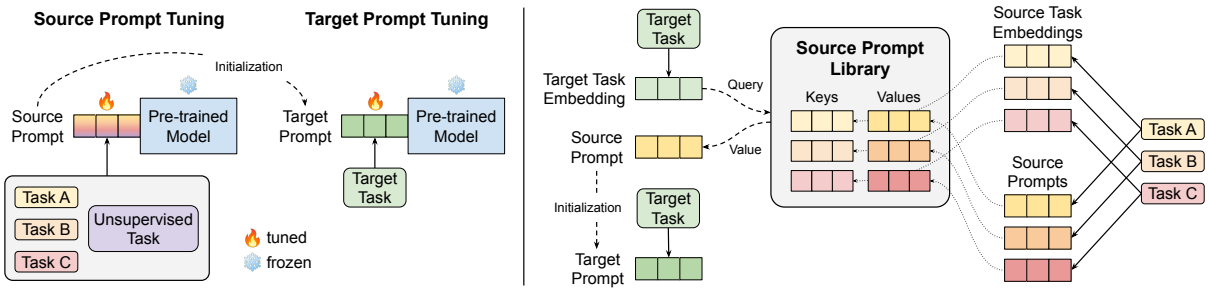

Figure 2: An illustration of our *generic* (left) and *targeted* (right) SPoT approaches. **Left**: We learn a single generic source prompt on one or more source tasks, which is then used to initialize the prompt for each target task. **Right**: We learn separate prompts for various source tasks, saving early checkpoints as task embeddings and best checkpoints as source prompts. These form the keys and values of our prompt library. Given a novel target task, a user: (i) computes a task embedding, (ii) retrieves an optimal source prompt, and (iii) trains a target prompt, initialized from the source prompt (see §3 for details).

that learns a small task-specific prompt (a sequence of tunable tokens prepended to each example) for each downstream task during adaptation to condition the frozen language model to perform the task. Strikingly, as model capacity increases, PROMPTTUNING becomes competitive with MODELTUNING, which fine-tunes the entire model on each downstream task. Nevertheless, at smaller model sizes (below 11B parameters), there are still large gaps between PROMPTTUNING and MODELTUNING.

In this paper, we propose SPoT: **S**oft **P**rompt **T**ransfer, a novel transfer learning approach in the context of prompt tuning. SPoT first trains a prompt on one or more source tasks, and then uses the resulting prompt to initialize the prompt for a target (downstream) task. Our experiments show that SPoT offers significant improvements over PROMPTTUNING across tasks and model sizes. For instance, on the SUPERGLUE benchmark (Wang et al., 2019b), we obtain +10.1 and +2.4 point average accuracy improvements using the T5 BASE (220M parameter) and T5 XXL (11B parameter) models (Raffel et al., 2020), respectively. More importantly, SPoT is competitive with or outperforms MODELTUNING across all model sizes (see Figure 1).

Motivated by these results, we investigate transferability between tasks, through the lens of soft task prompts. Our goal is to answer two questions: (a) *For a given target task, when does initializing the prompt from a source task boost performance?* (b) *Can we use task prompts to efficiently predict which source tasks will transfer well onto a novel target task?* To answer (a), we conduct a systematic study of the T5 model using 26 NLP tasks in 160 combinations of source and target tasks. Our results indicate that many tasks can benefit each other via prompt transfer. To address (b), we interpret the learned task prompts as *task embeddings* to construct a semantic space of tasks and formalize the similarity between tasks. We design an efficient retrieval algorithm that measures task embedding similarity, allowing practitioners to identify source tasks that will likely yield positive transfer.

To summarize, our main contributions are: (1) We propose SPoT, a novel prompt-based transfer learning approach, and show that scale is not necessary for PROMPTTUNING to match the performance of MODELTUNING; on SUPERGLUE, SPoT matches or beats MODELTUNING across all model sizes. (2) We conduct a large-scale and systematic study on task transferability, demonstrating conditions under which tasks can benefit each other via prompt transfer. (3) We propose an efficient retrieval method that interprets task prompts as task embeddings to construct a semantic space of tasks, and measures task embedding similarity to identify which tasks could benefit each other. (4) To facilitate future work on prompt-based learning, we will release our library of task prompts and pre-trained models, and provide practical recommendations for adapting our library to NLP practitioners at https://github.com/google-research/prompt-tuning/tree/main/prompt_tuning/spot.

2 Improving PROMPTTUNING with SPoT

To improve performance of PROMPTTUNING on a target task, SPoT introduces *source prompt tuning*, an intermediate training stage between language model pre-training and target prompt tuning (Figure 2, left), to learn a prompt on one or more source tasks (while still keeping the base model frozen),

which is then used to initialize the prompt for the target task.[1] Our approach retains all the computational benefits of PROMPTTUNING: for each target task, it only requires storing a small task-specific prompt, enabling the reuse of a single frozen pre-trained model across all tasks. In this section, we present a *generic* SPOT approach where a single transferred prompt is reused for all target tasks. In §3, we explore a *targeted* approach that retrieves different source prompts for different target tasks.

2.1 Experimental setup

Our frozen models are built on top of the pre-trained T5 checkpoints of all sizes: SMALL, BASE, LARGE, XL, XXL with 60M, 220M, 770M, 3B, and 11B parameters, respectively. In our experiments with SPOT, we leverage the LM adapted version of T5[2], which was found to be easier to optimize for PROMPTTUNING (Lester et al., 2021).

2.1.1 Baselines

We compare SPOT to the following baselines:

PROMPTTUNING: The vanilla prompt tuning approach of Lester et al. (2021), where an independent prompt is directly trained on each target task.

MODELTUNING & MULTI-TASKMODELTUNING: We compare prompt tuning approaches to MODELTUNING, the standard fine-tuning approach (Devlin et al., 2019; Raffel et al., 2020), where all model parameters are fine-tuned on each target task separately. For an apples-to-apples comparison, we include MULTI-TASKMODELTUNING, a more competitive baseline that first fine-tunes the entire model on the same mixture of source tasks used for SPOT before fine-tuning it on individual target tasks.[3]

2.1.2 Evaluation datasets

We study downstream performance on a diverse set of tasks from the GLUE (Wang et al., 2019c) and SUPERGLUE (Wang et al., 2019b) benchmarks.[4] We train for a fixed number of steps and report results on the validation set associated with each dataset.[5]

2.1.3 Data for source prompt tuning

As with language model pre-training, the choice of training data is crucial for successful prompt transfer. To investigate the impact of source training data on downstream performance, we compare a diverse set of source tasks.

A single unsupervised learning task: We first consider training the prompt on a fraction of the C4 (Colossal Clean Crawled Corpus) dataset (Raffel et al., 2020) using the "prefix LM" objective discussed in Raffel et al. (2020). Although this task was used to pre-train our frozen T5 models already, it could still be helpful for learning a general-purpose prompt.

A single supervised learning task: Alternatively, we can train the prompt using a supervised task. We use either MNLI (Williams et al., 2018) or SQUAD (Rajpurkar et al., 2016) as a single source task. MNLI was shown to be helpful for many sentence-level classification tasks (Phang et al., 2019), while SQUAD was found to generalize well to QA tasks (Talmor and Berant, 2019).

A multi-task mixture: So far, we have considered using a single source task. An alternative approach is multi-task training. Within T5's unified text-to-text framework, this simply corresponds to mixing different datasets together. We explore mixing datasets from different NLP benchmarks or families of tasks, including GLUE, SUPERGLUE, natural language inference (NLI), paraphrasing/semantic similarity, sentiment analysis, question answering (QA) on MRQA (Fisch et al., 2019), commonsense reasoning on RAINBOW (Lourie et al., 2021), machine translation, summarization, and natural lan-

[1] The target task can be treated as one of the source tasks being mixed together.

[2] T5 1.1 checkpoints trained for an additional 100K steps using the "prefix LM" objective (Raffel et al., 2020), available at https://github.com/google-research/text-to-text-transfer-transformer/blob/main/released_checkpoints.md

[3] In preliminary experiments, we found that using the original version of T5 1.1 (which was pre-trained exclusively on span corruption) for model tuning approaches results in better performance than using the LM adapted version. We therefore report results corresponding to the original T5 1.1 for MODELTUNING and MULTI-TASKMODELTUNING.

[4] These datasets include grammatical acceptability judgments (COLA (Warstadt et al., 2019)), sentiment analysis (SST-2 (Socher et al., 2013)), paraphrasing/semantic similarity (MRPC (Dolan and Brockett, 2005), STS-B (Cer et al., 2017), QQP (Iyer et al., 2017)), natural language inference (MNLI (Williams et al., 2018), QNLI (Wang et al., 2019c), RTE (Dagan et al., 2005, et seq.), CB (De Marneffe et al., 2019)), coreference resolution (WSC (Levesque et al., 2012)), sentence completion (COPA (Roemmele et al., 2011)), word sense disambiguation (WIC (Pilehvar and Camacho-Collados, 2019)), and question answering (MULTIRC (Khashabi et al., 2018), RECORD (Zhang et al., 2018), BOOLQ (Clark et al., 2019)). We exclude the problematic WNLI (Levesque et al., 2012) dataset from GLUE, following Devlin et al. (2019).

[5] For tasks with multiple metrics, we average the metrics.

guage generation on GEM (Gehrmann et al., 2021).[6] We create a mixture of source tasks from each of the NLP benchmarks/families of tasks above, and a mixture comprising all datasets (C4 + 55 labeled datasets), using the examples-proportional mixing strategy in Raffel et al. (2020) with an artificial dataset size limit $\mathcal{K} = 2^{19}$ examples.

2.1.4 Training details

We closely follow the training procedure in Lester et al. (2021). Specifically, the only new parameters introduced during both source and target prompt tuning are a shared prompt $\boldsymbol{\rho} \in \mathbb{R}^{\mathcal{L} \times \mathcal{E}}$ prepended to each (embedded) input sequence, where $\mathcal{L}$, $\mathcal{E}$ are the prompt length and the embedding size, respectively. In all cases, we set $\mathcal{L} = 100$ tokens and tune the prompt for a fixed number of steps $\mathcal{S}$.[7] While $\mathcal{S}$ is set to 30K in Lester et al. (2021), we find that additional tuning is helpful on large datasets. As such, we set $\mathcal{S}$ to $2^{18} = 262{,}144$, following Raffel et al. (2020), with the exception of ablation experiments (rows "− longer tuning") in Table 1 which use $\mathcal{S} = 30$K. For source prompt tuning, the prompt token embeddings are initialized from sampled vocabulary (i.e., the 5,000 most common tokens). During target prompt tuning, we save a checkpoint every 500 steps and report results on the checkpoint with the highest validation performance. Appendix C contains training details for PROMPTTUNING and model tuning approaches.

2.2 Effect of SPOT

We compare the results of SPOT and other approaches in Table 1 and Figure 1. Below, we summarize and analyze each of our findings in detail.

SPOT significantly improves performance and stability of PROMPTTUNING: Our results on the GLUE and SUPERGLUE benchmarks with T5 BASE (Table 1) suggest that prompt transfer provides an effective means of improving performance for PROMPTTUNING. For example, the best-performing variant of SPOT outperforms the vanilla PROMPTTUNING approach on both GLUE and SUPERGLUE by a substantial margin, obtaining +4.4 and +10.1 point average accuracy improvements, respectively. Our ablation study indicates that longer tuning is also an important ingredient for achieving the best performance, and is complementary to prompt transfer. Additionally, when longer tuning is omitted, we observe that SPOT improves stability across runs.

Method	GLUE	SUPERGLUE
BASELINE		
PROMPTTUNING	$81.2_{0.4}$	$66.6_{0.2}$
− longer tuning	$78.4_{1.7}$	$63.1_{1.1}$
SPOT *with different source mixtures*		
GLUE (*8 tasks*)	$\mathbf{82.8}_{0.2}$	$\mathbf{73.2}_{0.3}$
− longer tuning	$82.0_{0.2}$	$70.7_{0.4}$
C4	$82.0_{0.2}$	$67.7_{0.3}$
MNLI	$82.5_{0.0}$	$72.6_{0.8}$
SQUAD	$82.2_{0.1}$	$72.0_{0.4}$
SUPERGLUE (8 tasks)	$82.0_{0.1}$	$66.6_{0.2}$
NLI (*7 tasks*)	$82.6_{0.1}$	$71.4_{0.2}$
Paraphrasing/similarity (*4 tasks*)	$82.2_{0.1}$	$69.7_{0.5}$
Sentiment (*5 tasks*)	$81.1_{0.2}$	$68.6_{0.1}$
MRQA (*6 tasks*)	$81.8_{0.2}$	$68.4_{0.2}$
RAINBOW (*6 tasks*)	$80.3_{0.6}$	$64.0_{0.4}$
Translation (*3 tasks*)	$82.4_{0.2}$	$65.3_{0.1}$
Summarization (*9 tasks*)	$80.9_{0.3}$	$67.1_{1.0}$
GEM (*8 tasks*)	$81.9_{0.2}$	$70.5_{0.5}$
All (C4 + 55 supervised tasks)	$81.8_{0.2}$	$67.9_{0.9}$

Table 1: GLUE and SUPERGLUE results achieved by applying T5 BASE with different prompt tuning approaches. We report the mean and standard deviation (in the subscript) across three random seeds. SPOT significantly improves performance and stability of PROMPTTUNING across the two benchmarks.

Within SPOT, we can compare the effectiveness of different source mixtures (see Table 1). Source prompt tuning on GLUE performs best on both GLUE and SUPERGLUE, obtaining average scores of 82.8 and 73.2, respectively.[8] Interestingly, unsupervised source prompt tuning on C4 (the same task used to pre-train our frozen models) still yields considerable improvements, even outperforming using SUPERGLUE for SUPERGLUE tasks. Using MNLI or SQUAD as a single source dataset is also particularly helpful across target tasks. Other source mixtures can lead to significant gains, with some families of tasks (e.g., NLI and paraphrasing/semantic similarity) showing more benefit than others. Mixing all the datasets together does not yield the best results, possibly due to task interference/negative transfer issues, where achieving good performance on one or more source tasks can hurt performance on a target task.

[6] See Appendix B for details about datasets.

[7] We use the Adafactor optimizer (Shazeer and Stern, 2018) with default parameters except with a constant learning rate of 0.3, weight decay of $1e{-}5$, and parameter scaling turned off. We train with a batch size of 32. The dropout probability is always kept at 0.1. All of our models are implemented using JAX (Bradbury et al., 2018) and FLAX (Heek et al., 2020).

[8] SUPERGLUE tasks benefit less from source prompt tuning on SUPERGLUE likely due to the small size of these datasets.

SPoT helps close the gap with MODELTUNING across all model sizes: Figure 1 shows our SUPERGLUE results across model sizes (see Appendix A for full results). As shown in Lester et al. (2021), PROMPTTUNING becomes more competitive with scale, and at the XXL size, it nearly matches the performance of MODELTUNING. However, at smaller model sizes, there are still large gaps between the two approaches. We show that SPOT helps close these gaps and even exceeds MODELTUNING's performance by a large margin at several model sizes, while retaining all the computational benefits conferred by PROMPTTUNING. Finally, at the XXL size, SPOT achieves the best average score of 91.2, +1.1 points better than the strong MULTI-TASKMODELTUNING baseline, despite having 27,000× fewer task-specific parameters.

As a final test of SPOT's effectiveness, we submitted our XXL model's predictions to the SUPERGLUE leaderboard, achieving a score of 89.2. This far exceeds all previous submissions using parameter-efficient adaptation, such as GPT-3 (71.8), and almost matches fully fine-tuned T5 XXL (89.3),[9] despite tuning 27,000× fewer parameters. To the best of our knowledge, SPOT is the first parameter-efficient adaptation approach that is competitive with methods that tune billions of parameters. See Appendix D for details.

3 Predicting task transferability

So far, we have seen that soft prompt transfer can significantly boost the performance of prompt tuning, but it is critical to pick the right source tasks for transfer. For instance, through an extensive search, we found that GLUE and MNLI provide excellent source tasks for transferring to individual GLUE and SUPERGLUE tasks. But what about a resource-constrained scenario where a user is not able to exhaustively search over a set of source tasks? Can we *predict* which tasks will best transfer onto a novel target task without testing them one by one?

To investigate this, we conduct a large-scale empirical study with 26 NLP tasks. We first measure transferability across all task combinations (§3.1). Next, we show that by interpreting task prompts as task embeddings, we can construct a semantic space of tasks, wherein similar tasks cluster together (§3.2). Based on this observation, we propose a retrieval algorithm (§3.3) that leverages task embedding similarity to choose which source tasks to use for a given novel target task (Figure 2, right). Our proposed approach can eliminate 69% of the source task search space while keeping 90% of the best-case quality gain.

[9]Note that the T5 submission uses the original version of T5 (which was pre-trained on a multi-task mixture of unsupervised and supervised tasks) while we use T5 1.1 (which was pre-trained on C4 only without mixing in supervised tasks).

Name	Task type	|Train|
16 source tasks		
C4	language modeling	365M
DOCNLI	NLI	942K
YELP-2	sentiment analysis	560K
MNLI	NLI	393K
QQP	paraphrase detection	364K
QNLI	NLI	105K
RECORD	QA	101K
CXC	semantic similarity	88K
SQUAD	QA	88K
DROP	QA	77K
SST-2	sentiment analysis	67K
WINOGRANDE	commonsense reasoning	40K
HELLASWAG	commonsense reasoning	40K
MULTIRC	QA	27K
COSMOSQA	commonsense reasoning	25K
RACE	QA	25K
10 target tasks		
BOOLQ	QA	9K
COLA	grammatical acceptability	9K
STS-B	semantic similarity	6K
WIC	word sense disambiguation	5K
CR	sentiment analysis	4K
MRPC	paraphrase detection	4K
RTE	NLI	2K
WSC	coreference resolution	554
COPA	QA	400
CB	NLI	250

Table 2: Tasks used in our task transferability experiments, sorted by training dataset size.

3.1 Measuring transferability

We study a diverse set of 16 source datasets and 10 target datasets (see Table 2).[10] We consider all 160 possible source-target pairs, and perform transfer from each source task to each target task. All source tasks are data-rich or have been shown to yield positive transfer in prior work. To simulate a realistic scenario, we use low-resource tasks (less than 10K training examples) as target tasks.[11]

[10]Beyond the datasets from §2, we use DOCNLI (Yin et al., 2021), YELP-2 (Zhang et al., 2015), CXC (Parekh et al., 2021), DROP (Dua et al., 2019), WINOGRANDE (Sakaguchi et al., 2020), HELLASWAG (Zellers et al., 2019), COSMOSQA (Huang et al., 2019), RACE (Lai et al., 2017), and CR (Hu and Liu, 2004).

[11]The source tasks comprise one unsupervised task (C4) and 15 supervised tasks covering natural language inference (NLI), paraphrasing/semantic similarity, sentiment analysis, question answering (QA), and commonsense reasoning. The target tasks additionally include grammatical acceptability, word sense disambiguation, and coreference resolution.

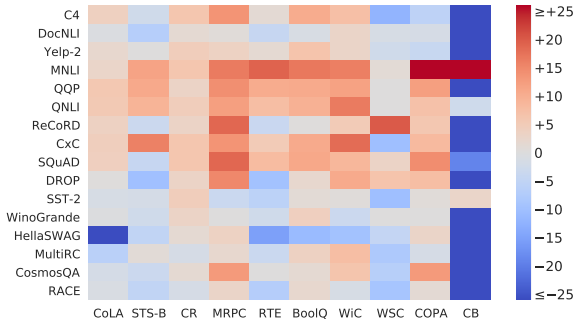

Figure 3: A heatmap of our task transferability results. Each cell shows the relative error reduction on the target task of the transferred prompt from the associated source task (row) to the associated target task (column).

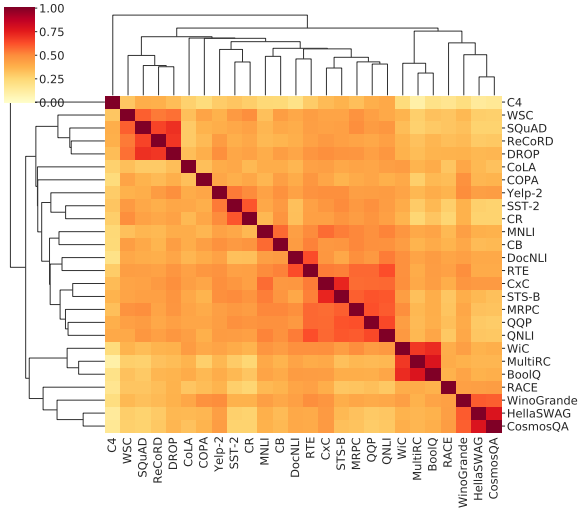

Figure 4: A clustered heatmap of cosine similarities between the task embeddings of the 26 NLP tasks we study. Our prompt-based task embeddings capture task relationships: similar tasks cluster together.

To limit computational costs, we use T5 BASE in all of our task transferability experiments. We perform 262,144 prompt tuning steps on each source task. The prompt checkpoint with the highest source task validation performance is selected to initialize prompts for target tasks. Since the target datasets are small, we only perform 100K prompt tuning steps on each target task. We repeat each experiment three times with different random seeds. Other training details match §2.1.4.

Tasks benefiting each other via prompt transfer: Figure 3 shows a heatmap of our results (see Appendix E for full results). In many cases, prompt transfer provides a significant gain on the target task. The transfer MNLI → CB yields the largest relative error reduction of 58.9% (from an average score of 92.7 to 97.0), followed by MNLI → COPA (29.1%) and RECORD → WSC (20.0%). Using the best source prompt (out of 48) for each target task dramatically improves the average score across our 10 target tasks from 74.7 to 80.7. Overall, our results show effective transfer from large source tasks that involve high-level reasoning about semantic relationships among sentences (e.g., MNLI), or when the source and target tasks are similar (e.g., CXC → STS-B). Interestingly, positive transfer can occur between relatively dissimilar tasks (e.g., RECORD → WSC, SQUAD → MRPC, CXC → WIC).[12]

3.2 Defining task similarity through prompts

Since only prompt parameters are updated during prompt tuning on specific tasks, the learned prompts likely encode task-specific knowledge. This suggests that they could be used to reason about the nature of tasks and their relationships. To test this idea, we interpret task prompts as *task embeddings* and construct a semantic space of tasks. More concretely, we define a task's embedding as the prompt checkpoint after training for 10K steps on that task.[13] Note that using early checkpoints allows for quick computation of task embeddings for novel target tasks. We estimate the similarity between two tasks t^1, t^2 by measuring the similarity between their corresponding task embeddings $\boldsymbol{e}^1, \boldsymbol{e}^2$, using the following metrics:

COSINE SIMILARITY OF AVERAGE TOKENS: We compute the cosine similarity between the average pooled representations of the prompt tokens:

$$sim(t^1, t^2) = cos(\frac{1}{\mathcal{L}}\sum_i \boldsymbol{e}_i^1, \frac{1}{\mathcal{L}}\sum_j \boldsymbol{e}_j^2),$$

where $\boldsymbol{e}_i^1, \boldsymbol{e}_j^2$ denote the respective prompt tokens of $\boldsymbol{e}^1, \boldsymbol{e}^2$, and cos denotes the cosine similarity.

PER-TOKEN AVERAGE COSINE SIMILARITY: We compute the average cosine similarity between every prompt token pair $(\boldsymbol{e}_i^1, \boldsymbol{e}_j^2)$:

$$sim(t^1, t^2) = \frac{1}{\mathcal{L}^2}\sum_i\sum_j cos(\boldsymbol{e}_i^1, \boldsymbol{e}_j^2).$$

[12] Table 7 in Appendix E contains more cases.

[13] Our preliminary experiments with other checkpoint alternatives (in the range 1K to 100K) yielded worse performance. We also found that measuring task similarity using task embeddings derived from a *fixed* prompt checkpoint (10K steps) gave better results than those derived from the *best-performing* prompt checkpoint per task. This suggests that prompts trained for a differing number of steps may be less directly comparable than those trained for the same duration.

Task embeddings capture task relationships: Figure 4 shows a hierarchically-clustered heatmap of cosine similarities between the task embeddings using the COSINE SIMILARITY OF AVERAGE TOKENS metric.[14] We observe that our learned task embeddings capture many intuitive task relationships. Specifically, similar tasks group together into clusters, including QA (SQUAD, RECORD, and DROP; MULTIRC and BOOLQ), sentiment analysis (YELP-2, SST-2, and CR), NLI (MNLI and CB; DOC-NLI and RTE), semantic similarity (STS-B and CXC), paraphrasing (MRPC and QQP), and commonsense reasoning (WINOGRANDE, HELLASWAG, and COSMOSQA). We note that QNLI, which is an NLI task built from the SQUAD dataset, is not closely linked to SQUAD; this suggests that our task embeddings are more sensitive to the type of task than domain similarity. Interestingly, they also capture the unintuitive case of RECORD's high transferability to WSC. Additionally, task embeddings that are derived from different prompts of the same task have high similarity scores (see Appendix F).

3.3 Predicting transferability via similarity

We leverage our task embeddings to predict and exploit task transferability. Specifically, we explore methods to predict the most beneficial source tasks for a given target task and then make use of the source task prompts to improve performance on the target task. To enlarge our set of source prompts, we use the prompts from each of the three different prompt tuning runs on each source task, resulting in 48 source prompts. Given a target task t with task embedding $\boldsymbol{e}^t$, we rank all the source prompts $\boldsymbol{\rho}^s$ with associated embeddings $\boldsymbol{e}^s$ in descending order by similarity, $sim(\boldsymbol{e}^s, \boldsymbol{e}^t)$. We denote the ranked list of source prompts as $\boldsymbol{\rho}^{s_r}$, where r denotes the rank ($r = 1, 2, \ldots, 48$). We experiment with three methods for using the ranked source prompts:

BEST OF TOP-k: We select the top-k source prompts and use each of them individually to initialize the target prompt. This procedure requires prompt tuning k times on the target task t. The best individual result is used for evaluating the effectiveness of this method.

TOP-k WEIGHTED AVERAGE: We initialize the target prompt with a weighted average of the top-k source prompts $\sum_{r=1}^{k} \alpha_r \boldsymbol{\rho}^{s_r}$ so that we only perform prompt tuning on the target task t once. The weights α_r are computed as:

$$\alpha_r = \frac{sim(\boldsymbol{e}^{s_r}, \boldsymbol{e}^t)}{\sum_{l=1}^{k} sim(\boldsymbol{e}^{s_l}, \boldsymbol{e}^t)},$$

where $\boldsymbol{e}^{s_r}$ denotes the corresponding task embedding of $\boldsymbol{\rho}^{s_r}$.

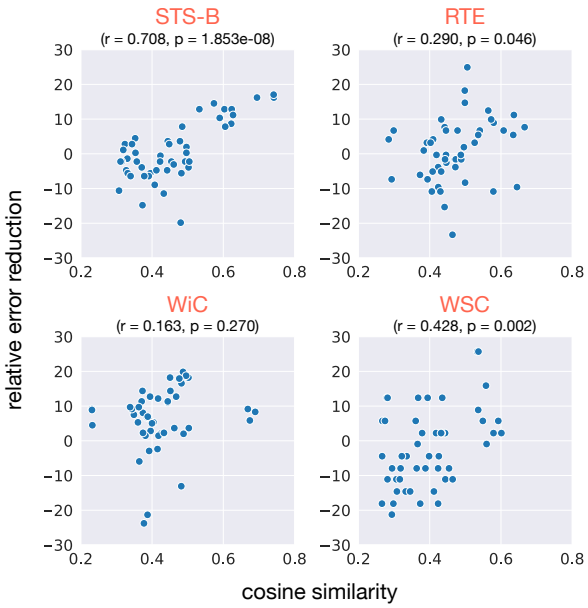

Figure 5: Correlation between task similarity and task transferability. Each point represents a source prompt. The x-axis shows the cosine similarity between the associated source and target task embeddings, averaged over three runs for the target task (orange title). The y-axis measures the relative error reduction on the target task achieved by each source prompt. We include the Pearson correlation coefficient (r) and p-value.

TOP-k MULTI-TASK MIXTURE: We first identify the source tasks whose prompts are in the top-k prompts and mix their datasets and the target dataset together, using the examples-proportional mixing strategy of Raffel et al. (2020). Then, we perform source prompt tuning on this multi-task mixture and use the final prompt checkpoint to initialize the target prompt.

We report the average score across all target tasks achieved by each method. For comparison, we measure the absolute and relative improvements over BASELINE—prompt tuning on each target task from scratch (i.e., without any prompt transfer).[15] Additionally, we include ORACLE—the oracle results achieved by a brute-force search to identify

[14] To obtain the highest resolution of similarity between two tasks, we use the average of cosine similarities between their task embeddings obtained with all the three different prompt tuning runs (9 combinations).

[15] For each target task t, we report the average and standard deviation of performance across three prompt tuning runs.

the best possible out of 48 source prompts for each target task.

Correlation between task similarity and task transferability: Figure 5 shows how the relative error reduction on a target task changes as a function of the similarity between the source and target task embeddings. Overall, we observe a significant positive correlation between task embedding similarity and task transferability on four (out of 10) target tasks, including STS-B ($p < 0.001$), CB ($p < 0.001$), WSC ($p < 0.01$), and RTE ($p < 0.05$), while it is less significant on the other tasks.[16] In some cases (e.g., on BOOLQ), we observe a large relative error reduction (19.0%, achieved by a source prompt of MNLI) despite a low cosine similarity (0.4). This suggests that factors other than task similarity (data size, task difficulty, domain similarity, etc.) may also play a role in determining transferability.

Retrieving targeted source tasks via task embeddings is helpful: Table 3 compares different methods for identifying which source prompts could be beneficial for a given target task. Overall, our results show the effectiveness of BEST OF TOP-k. Simply choosing the source prompt with the highest task embedding similarity to the target task using PER-TOKEN AVERAGE COSINE SIMILARITY improves over the baseline by a large margin (from an average score of 74.7 to 76.7, a 12.1% average relative error reduction). Trying all the top-3 (out of 48) source prompts for each target task yields an average score of 77.5. With larger values of k, we can retain most of the benefits of oracle selection (80% of the gain in terms of average score with $k = 9$ and 90% with $k = 15$), while still eliminating over 2/3 of the candidate source prompts. TOP-k WEIGHTED AVERAGE has similar average performance to BEST OF TOP-k with $k = 1$, but achieves lower variance. Thus, this may be an appealing alternative to BEST OF TOP-k in scenarios where trying multiple prompt tuning runs on the target task is computationally prohibitive. Finally, TOP-k MULTI-TASK MIXTURE also provides a means of obtaining strong performance with an average score of 77.8, even outperforming BEST OF TOP-k with $k \leq 3$.

Method	Change		Avg. score
	Abs.	Rel.	
BASELINE	-	-	$74.7_{0.7}$
BRUTE-FORCE SEARCH ($k = 48$)			
ORACLE	$6.0_{0.5}$	$26.5_{1.1}$	$80.7_{0.0}$
COSINE SIMILARITY OF AVERAGE TOKENS			
BEST OF TOP-k			
$k = 1$	$1.5_{0.5}$	$11.7_{1.1}$	$76.2_{0.1}$
$k = 3$	$2.7_{0.6}$	$16.6_{1.1}$	$77.4_{0.3}$
$k = 6$	$3.8_{0.1}$	$20.0_{1.1}$	$78.5_{0.5}$
$k = 9$	$4.5_{0.4}$	$22.2_{1.1}$	$79.2_{0.1}$
$k = 12$	$5.0_{0.9}$	$23.6_{2.2}$	$79.7_{0.4}$
$k = 15$	$5.4_{0.8}$	$24.9_{1.8}$	$80.1_{0.3}$
PER-TOKEN AVERAGE COSINE SIMILARITY			
BEST OF TOP-k			
$k = 1$	$2.0_{0.4}$	$12.1_{1.1}$	$76.7_{0.7}$
$k = 3$	$2.9_{0.6}$	$17.0_{0.6}$	$77.5_{0.4}$
$k = 6$	$4.5_{0.5}$	$22.1_{1.2}$	$79.2_{0.1}$
$k = 9$	$4.6_{0.5}$	$22.6_{0.9}$	$79.5_{0.2}$
$k = 12$	$5.0_{0.6}$	$23.5_{1.4}$	$79.6_{0.1}$
$k = 15$	$5.3_{0.9}$	$24.5_{2.2}$	$80.0_{0.4}$
TOP-k WEIGHTED AVERAGE			
best $k = 3$	$1.9_{0.5}$	$11.5_{2.7}$	$76.6_{0.1}$
TOP-k MULTI-TASK MIXTURE			
best $k = 12$	$3.1_{0.5}$	$15.3_{2.8}$	$77.8_{0.1}$

Table 3: Task embeddings provide an effective means of predicting and exploiting task transferability. Using BEST OF TOP-k with $k = 3$ improves over BASELINE (PROMPTTUNING on each task from scratch) by +2.8 points. With larger values of k (≤ 15), we can retain most of the benefits conferred by oracle selection. For TOP-k WEIGHTED AVERAGE and TOP-k MULTI-TASK MIXTURE, we experiment with different values of $k \in \{3, 6, 9, 12\}$ and report the best results.

4 Related Work

Parameter-efficient transfer learning: Large-scale pre-trained language models have been shown to exhibit remarkable performance on many NLP tasks (Devlin et al., 2019; Liu et al., 2019b; Yang et al., 2019; Lan et al., 2020; Raffel et al., 2020; Brown et al., 2020; He et al., 2021). To improve practical applicability of these models, early work introduces compression techniques (Sanh et al., 2019; Jiao et al., 2020; Fan et al., 2020; Sanh et al., 2020) to obtain lightweight models. Other work explores updating only small parts of the model (Zaken et al., 2021) or task-specific modules, such as adapters (Houlsby et al., 2019; Karimi Mahabadi et al., 2021) or low-rank structures (Mahabadi et al., 2021; Hu et al., 2021), while keeping the rest of the model fixed.

Recently, Brown et al. (2020) demonstrate impressive few-shot performance with PROMPTDESIGN, where their model is conditioned on a manual text prompt at inference time to perform different tasks. Several efforts have since focused on developing prompt-based learning approaches with carefully handcrafted prompts (Schick and Schütze, 2021), prompt mining and paraphrasing (Jiang

[16] See Appendix G for full results.

et al., 2020b), gradient-based search for improved prompts (Shin et al., 2020), and automatic prompt generation (Gao et al., 2021). The use of hard prompts, however, was found to be sub-optimal and sensitive to the choice of the prompt (Zhao et al., 2021; Liu et al., 2021b). As such, more recent work has shifted toward learning soft prompts (Liu et al., 2021b; Qin and Eisner, 2021; Li and Liang, 2021; Lester et al., 2021), which can be seen as learnable parameters injected into the model. We refer readers to Liu et al. (2021a) for a recent survey on prompt-based learning research.

In concurrent work, Gu et al. (2021) also explore the effectiveness of prompt transfer. Their method uses hand-crafted pre-training tasks tailored to specific types of downstream tasks, being less extensible to novel downstream tasks. In contrast, we use existing tasks as source tasks and show that prompt transfer can confer benefits even when there are mismatches (e.g., in task type or input/output format) between the source and target.

Task transferability We also build on existing work on task transferability (Wang et al., 2019a; Liu et al., 2019a; Talmor and Berant, 2019; Pruksachatkun et al., 2020; Vu et al., 2020, 2021). Prior work shows effective transfer from data-rich source tasks (Phang et al., 2019), those that require complex reasoning and inference (Pruksachatkun et al., 2020), or those that are similar to the target task (Vu et al., 2020). There have also been efforts to predict task transferability (Bingel and Søgaard, 2017; Vu et al., 2020; Poth et al., 2021). Vu et al. (2020) use task embeddings derived from either the input text or the diagonal Fisher information matrix of the model, while Poth et al. (2021) explore adapter-based alternatives. Here, our use of the same model (without task-specific components) with a unifying text-to-text format allows us to more easily model the space of tasks. Additionally, prompt-based task embeddings are comparatively cheaper to obtain.

5 Limitations & Future work

As other parameter-efficient adaptation methods (see §4) may outperform PROMPTTUNING in specific situations, it would be interesting to test whether an approach similar to SPOT could extend successfully to these methods. At the same time, we believe that PROMPTTUNING has its own merits. As pre-trained language models become larger and larger, some advantages of PROMPTTUNING over other methods are: (1) Among current methods with learnable parameters, PROMPTTUNING is the most parameter efficient, requiring less than 0.01% task-specific parameters for most model sizes. (2) PROMPTTUNING is simpler than other methods, as it does not modify the internal model architecture (cf. the PREFIX-TUNING method of Li and Liang (2021), which adds a prefix to each layer of both the Transformer encoder and decoder); as such, PROMPTTUNING allows mixed-task inference and facilitates transfer learning between tasks. (3) As model capacity increases, PROMPTTUNING becomes more competitive with MODELTUNING; to the best of our knowledge, this has not been shown for other methods. (4) Soft prompts could possibly be interpreted as natural language instructions.

Additionally, since our prompt-based task embedding approach does not capture all of the factors that influence task transferability, we leave further exploration of other task embedding methods to future work.

6 Conclusion

In this paper, we study transfer learning in the context of prompt tuning. We show that scale is not necessary for PROMPTTUNING to match the performance of MODELTUNING. On SUPERGLUE, our SPOT approach matches or even exceeds the performance of MODELTUNING by a large margin across model sizes while being more parameter-efficient. Our large-scale study on task transferability indicates that tasks can benefit each other via prompt transfer in various scenarios. Finally, we demonstrate that task prompts can be interpreted as task embeddings to formalize the similarity between tasks. We propose a simple yet efficient retrieval approach that measures task similarity to identify which source tasks could confer benefits to a novel target task. Taken as a whole, we hope that our work will spur more research into prompt-based transfer learning.

Acknowledgements

We thank Mohit Iyyer, Sebastian Ruder, Kalpesh Krishna, Thang Luong, Quoc Le, and the members of the Descartes team and the UMass NLP group for helpful discussion and feedback. We would also like to thank Grady Simon, Lucas Dixon, Slav Petrov, Nader Akoury, Haw-Shiuan Chang, Katherine Thai, Marzena Karpinska, and Shufan Wang for their comments on this manuscript. Finally, we are grateful to Vamsi Aribandi for his work on preprocessing several datasets used in our experiments.

References

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 164–169.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2020)*, 34(05):7432–7439.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT 2014)*, pages 12–58.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation (WMT 2016)*, pages 131–198.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation (WMT 2015)*, pages 1–46.

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 (EMNLP 2015)*, pages 632–642.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: composable transformations of Python+NumPy programs.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, volume 33, pages 1877–1901.

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.

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 (ACL 2019)*, pages 2924–2936.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the 1st International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment (MLCW 2005)*, page 177–190.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The CommitmentBank: Investigating projection in naturally occurring discourse. In *Proceedings of Sinn und Bedeutung 23 (SuB 2018)*, volume 23, pages 107–124.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 4040–4054.

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 (NAACL 2019)*, pages 4171–4186.

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

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019)*, pages 2368–2378.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *Proceedings of the 12th International Conference on Natural Language Generation (INLG 2019)*, pages 421–426.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 1074–1084.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering (MRQA 2019)*, pages 1–13.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL 2021)*, pages 3816–3830.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 179–188.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization (NewSum 2019)*, pages 70–79.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2018)*, pages 708–719.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. PPT: Pre-trained prompt tuning for few-shot learning. *arXiv preprint arXiv:2109.04332*.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*, pages 4921–4933.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. 2020. Flax: A neural network library and ecosystem for JAX.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NeurIPS 2020)*, volume 28.

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 (PMLR 2019)*, volume 97, pages 2790–2799.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004)*, page 168–177.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 2391–2401.

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

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020a. Neural CRF model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 7943–7960.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020b. How can we know what language models know? *Transactions of the Association for Computational Linguistics (TACL 2020)*, 8:423–438.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics (Findings of EMNLP 2020)*, pages 4163–4174.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1601–1611.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*, pages 565–576.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2018)*, pages 252–262.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization (NewSum 2019)*, pages 48–56.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics (TACL 2019)*, 7:452–466.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics (Findings of EMNLP 2020)*, pages 4034–4048.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 785–794.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*.

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 (EMNLP 2021)*, pages 3045–3059.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In

Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2012), page 552–561.

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 (ACL 2021)*, pages 4582–4597.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics (Findings of EMNLP 2020)*, pages 1823–1840.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019)*, pages 1073–1094.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2021)*, 35(15):13480–13488.

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

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2021)*, pages 432–447.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 1797–1807.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 4885–4901.

Zarana Parekh, Jason Baldridge, Daniel Cer, Austin Waters, and Yinfei Yang. 2021. Crisscrossed captions: Extended intramodal and intermodal semantic similarity judgments for MS-COCO. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*, pages 2855–2870.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2019. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019)*, pages 1267–1273.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? Efficient intermediate task selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, pages 10585–10605.

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 language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 5231–5247.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2021)*, pages 5203–5212.

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 (JMLR 2020)*, 21(140):1–67.

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

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

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Proceedings of the 25th AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning (AAAI Spring Symposium 2011)*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 379–389.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2020)*, 34(05):8732–8740.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (EMC2 2019)*.

Victor Sanh, Thomas Wolf, and Alexander Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, volume 33, pages 20378–20389.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 4463–4473.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2021)*, pages 2339–2352.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1073–1083.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *arXiv preprint arXiv:1804.04235*.

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 2020)*, pages 4222–4235.

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 (EMNLP 2013)*, pages 1631–1642.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 4911–4921.

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 Workshop on Representation Learning for NLP (RepL4NLP 2017)*, pages 191–200.

Tu Vu, Minh-Thang Luong, Quoc Le, Grady Simon, and Mohit Iyyer. 2021. STraTA: Self-training with task augmentation for better few-shot learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, pages 5715–5731.

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 2020)*, pages 7882–7926.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. 2019a. Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling. In *Proceedings of the Annual Meeting of*

the Association for Computational Linguistics (ACL 2019), pages 4465–4476.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019b. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS 2019)*, volume 32, pages 3266–3280.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019c. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*.

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

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of the 33th Conference on Neural Information Processing Systems (NeurIPS 2019)*, volume 32, pages 5753–5763.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 2369–2380.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. DocNLI: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics (Findings of ACL-IJCNLP 2021)*, pages 4913–4922.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 4791–4800.

Rui Zhang and Joel Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 446–456.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NeurIPS 2015)*, volume 28, pages 649–657.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, volume 139 of *PMLR*, pages 12697–12706.

Appendices

A Full results for Figure 1

Table 4 shows the performance of different model tuning and prompt tuning methods (described in §2.1.1) on the SUPERGLUE benchmark.

Method	Model size				
	SMALL	BASE	LARGE	XL	XXL
PROMPTDESIGN (GPT-3)	40.6	43.4	45.1	47.8	52.8
MODELTUNING	$62.8_{0.8}$	$73.7_{0.6}$	$81.3_{0.6}$	$83.1_{0.2}$	$89.9_{0.2}$
PROMPTTUNING	$59.8_{0.8}$	$63.1_{1.1}$	$74.5_{2.2}$	$79.2_{0.9}$	$88.8_{0.2}$
MULTI-TASKMODELTUNING	$\mathbf{64.6}_{0.2}$	$\mathbf{79.2}_{0.3}$	$\mathbf{84.5}_{0.1}$	$88.0_{0.5}$	$90.1_{0.2}$
SPOT (OURS)	$64.5_{0.3}$	$73.2_{0.3}$	$82.7_{0.2}$	$\mathbf{88.7}_{0.3}$	$\mathbf{91.2}_{0.1}$

Table 4: SUPERGLUE performance of different model tuning and prompt tuning methods across model sizes. We report the mean and standard deviation (in the subscript) across three random seeds. SPOT outperforms vanilla PROMPTTUNING and GPT-3 by a large margin, matching or outperforming MODELTUNING across all model sizes. At the XXL model size, SPOT even outperforms MULTI-TASKMODELTUNING, which fine-tunes the entire model on the GLUE mixture before fine-tuning it on individual SUPERGLUE tasks.

B Source datasets used in our SPOT experiments in §2

Figure 6 displays the datasets used in our SPOT experiments in §2. In addition to the C4 unlabeled dataset (Raffel et al., 2020), we use 55 labeled datasets. These datasets come from common NLP benchmarks/families of tasks, namely:

- GLUE (Wang et al., 2019c), including COLA (Warstadt et al., 2019), SST-2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), QQP (Iyer et al., 2017), STS-B (Cer et al., 2017), MNLI (Williams et al., 2018), QNLI (Wang et al., 2019c), and RTE (Dagan et al., 2005, et seq.).
- SUPERGLUE (Wang et al., 2019b), including BOOLQ (Clark et al., 2019), CB (De Marneffe et al., 2019), COPA (Roemmele et al., 2011), MULTIRC (Khashabi et al., 2018), RECORD (Zhang et al., 2018), RTE, WIC (Pilehvar and Camacho-Collados, 2019), and WSC (Levesque et al., 2012).
- Natural language inference (NLI), including ANLI (Nie et al., 2020), CB, DOCNLI (Yin et al., 2021), MNLI, QNLI, RTE, and SNLI (Bowman et al., 2015).
- Paraphrasing/semantic similarity, including CXC (Parekh et al., 2021), MRPC, QQP, and STS-B.
- Sentiment analysis, including CR (Hu and Liu, 2004), GOEMOTIONS (Demszky et al., 2020), SENTIMENT140 (Go et al., 2009), SST-2, and YELP-2 (Zhang et al., 2015).
- Question answering (QA) on MRQA (Fisch et al., 2019), including SQUAD (Rajpurkar et al., 2016), NEWSQA (Trischler et al., 2017), TRIVIAQA (Joshi et al., 2017), SEARCHQA (Dunn et al., 2017), HOTPOTQA (Yang et al., 2018), and NATURALQUESTIONS (NQ (Kwiatkowski et al., 2019)).
- Commonsense reasoning on RAINBOW (Lourie et al., 2021) including αNLI (Bhagavatula et al., 2020), COSMOSQA (Huang et al., 2019), HELLASWAG (Zellers et al., 2019), PIQA (Bisk et al., 2020), SOCIALIQA (Sap et al., 2019), and WINOGRANDE (Sakaguchi et al., 2020).
- Machine translation, including WMT ENDE (Bojar et al., 2014), WMT ENFR (Bojar et al., 2015), and WMT ENRO (Bojar et al., 2016).
- Summarization, including AESLC (Zhang and Tetreault, 2019), BILLSUM (Kornilova and Eidelman, 2019), CNN/DAILYMAIL (Hermann et al., 2015; See et al., 2017), WIKILINGUA (Ladhak et al., 2020), GIGAWORD (Graff et al., 2003; Rush et al., 2015), MULTINEWS (Fabbri et al., 2019), NEWSROOM (Grusky et al., 2018), SAMSUM (Gliwa et al., 2019), and XSUM (Narayan et al., 2018).
- Natural language generation on GEM (Gehrmann et al., 2021), including COMMONGEN (Lin et al., 2020), DART (Nan et al., 2021), E2E (Dušek et al., 2019), SGD (Rastogi et al., 2020), WEBNLG (Gardent et al., 2017), WIKIAUTO (Jiang et al., 2020a), XSUM, and WIKILINGUA.

C Additional training details

For PROMPTTUNING, following Lester et al. (2021), we initialize the prompt tokens with embeddings that represent an enumeration of the output classes

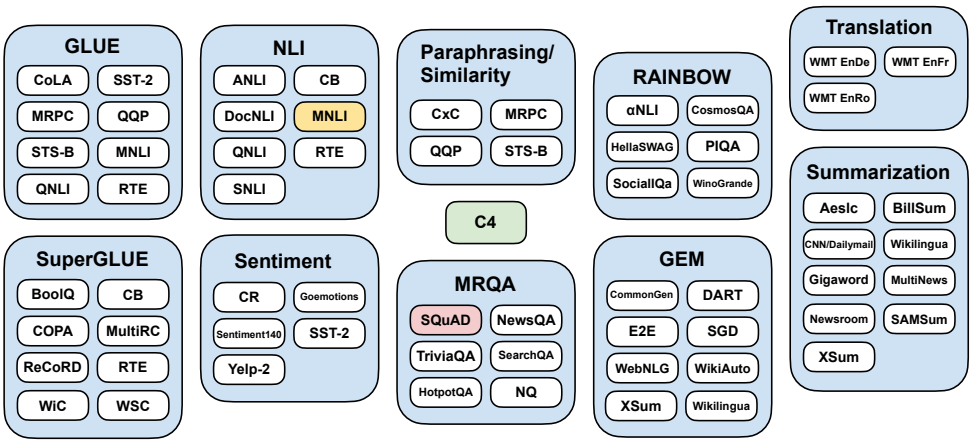

Figure 6: Datasets used in our SPoT experiments in §2. C4, MNLI, and SQuAD were all used by themselves as single source tasks in addition to being mixed in with other tasks.

	Model	Total parameters	Tuned parameters	Score	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC
Top-7 submissions	ST-MoE-32B	269B	269B	**91.2**	92.4	96.9/98.0	99.2	89.6/65.8	95.1/94.4	93.5	77.7	96.6
	Turing NLR v5	5.4B	5.4B	90.9	92.0	95.9/97.6	98.2	88.4/63.0	96.4/95.9	94.1	77.1	97.3
	ERNIE 3.0	12B	12B	90.6	91.0	98.6/99.2	97.4	88.6/63.2	94.7/94.2	92.6	77.4	97.3
	T5 + UDG	11B	11B	90.4	91.4	95.8/97.6	98.0	88.3/63.0	94.2/93.5	93.0	77.9	96.6
	DeBERTa / TuringNLRv4	3.1B	3.1B	90.3	90.4	95.7/97.6	98.4	88.2/63.7	94.5/94.1	93.2	77.5	95.9
	Human Baselines	-	-	89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0
	T5	11B	11B	89.3	91.2	93.9/96.8	94.8	88.1/63.3	94.1/93.4	92.5	76.9	93.8
Parameter-efficient adaptation	Frozen T5 1.1 + SPoT	11B	410K	**89.2**	91.1	95.8/97.6	95.6	87.9/61.9	93.3/92.4	92.9	75.8	93.8
	GPT-3 few-shot	175B	0	71.8	76.4	52.0/75.6	92.0	75.4/30.5	91.1/90.2	69.0	49.4	80.1
	WARP few-shot	223M	25K	48.7	62.2	70.2/82.4	51.6	0.0/0.5	14.0/13.6	69.1	53.1	63.7
	CBoW	15M	33K	44.5	62.2	49.0/71.2	51.6	0.0/0.5	14.0/13.6	49.7	53.1	65.1

Table 5: SuperGLUE results of our SPoT XXL submission (in green) and competitors from the leaderboard as of 2022/02/09.

with a back off to sampled vocabulary to fill any remaining prompt positions.

For model tuning approaches, we use the default hyperparameters for T5 (Raffel et al., 2020), i.e., learning rate 0.001, Adafactor optimizer with pre-training parameter states restored, and dropout probability 0.1. To improve the model tuning baselines, we perform a sweep over the batch size hyperparameter and select 2^{16} tokens per batch, following Lester et al. (2021).

D Details of our SuperGLUE submission

Table 5 shows the performance of our SPoT XXL SuperGLUE submission, along with several strong competitors from the public SuperGLUE leaderboard. Apart from the human baseline, the top-7 submissions all tune >3B parameters directly on the final tasks. Only three previous SuperGLUE submissions use *parameter efficient adaptation*, in the sense of tuning <1M parameters on the final tasks; all other submissions tune >50M parameters.[17]

Our SPoT submission achieves a score of 89.2, which far exceeds all other parameter-efficient adaptation methods, including GPT-3, which benefits from over 10× more frozen parameters (although it uses no tuned parameters). Compared to WARP (Hambardzumyan et al., 2021), our SPoT approach tunes 16× more parameters (410K vs. 25K), and benefits from 50× more frozen parameters.

To the best of our knowledge, SPoT is the first parameter-efficient adaptation approach that is competitive with methods that tune billions of parameters. Most notably, SPoT's performance almost matches that of fully fine-tuned T5 XXL (89.3), despite building on the same underlying model, and tuning 27,000× fewer parameters. We note that SPoT outperforms T5 on three of eight SuperGLUE tasks (namely, CB, COPA, RTE).

E Task transferability results

The full results of our task transferability experiments can be found in Table 6. We show that in many cases, initializing the prompt to that of a

[17] The "AILabs Team, Transformers" submission is listed as tuning 3M parameters, but we suspect this is in error, as the submission mentions using the T5-3B and T5-Large models.

source task can provide significant gain on a target task. Table 7 displays positive transfers with more than 10% relative error reduction on the target task.

F Task embedding similarity

In Figure 7, we show a clustered heatmap of cosine similarities between the task embeddings of the 26 NLP tasks we study in our task transferability experiments. For each task, we include the resulting task embeddings from all the three different prompt tuning runs on the task. As can be seen, our task embeddings capture task relationships: similar tasks cluster together. Additionally, task embeddings that are derived from different prompts of the same task are linked together.

G Correlation between task similarity and task transferability

Figure 8 shows how the relative error reduction on a target task changes as a function of the similarity between the source and target task embeddings.

	BoolQ	CoLA	STS-B	WiC	CR	MRPC	RTE	WSC	COPA	CB
Baseline	$73.0_{1.2}$	$52.9_{1.2}$	$88.1_{0.6}$	$63.6_{1.6}$	$93.5_{0.2}$	$86.1_{0.7}$	$68.7_{1.2}$	$71.5_{1.7}$	$56.7_{1.7}$	$92.7_{1.9}$
C4	$75.8_{0.5}$	$54.8_{1.1}$	$87.8_{0.6}$	$66.3_{0.8}$	**93.9**$_{0.1}$	$88.0_{0.6}$	$69.1_{1.9}$	$68.0_{0.5}$	$54.3_{0.9}$	$83.1_{5.7}$
DocNLI	$72.7_{1.4}$	$52.7_{0.9}$	$87.3_{0.9}$	$64.7_{0.3}$	$93.6_{0.4}$	$86.2_{0.8}$	$67.4_{2.6}$	$71.1_{3.6}$	$56.0_{5.9}$	$87.2_{1.7}$
Yelp-2	$74.8_{0.7}$	$53.9_{0.2}$	$88.1_{0.3}$	$64.7_{0.5}$	$93.8_{0.3}$	$86.6_{0.8}$	$69.2_{1.1}$	$70.8_{1.2}$	$55.0_{0.0}$	$87.8_{1.6}$
MNLI	**77.6**$_{0.4}$	$54.2_{0.7}$	$89.5_{0.3}$	$69.5_{0.5}$	**93.9**$_{0.4}$	$88.4_{0.6}$	**74.7**$_{1.3}$	$71.8_{3.3}$	**69.3**$_{2.1}$	**97.0**$_{1.1}$
QQP	$75.9_{0.5}$	**55.6**$_{1.3}$	$89.4_{0.2}$	$67.9_{0.2}$	$93.7_{0.5}$	$88.1_{0.7}$	$72.0_{0.5}$	$71.5_{0.9}$	$62.0_{2.2}$	$88.7_{4.2}$
QNLI	$75.6_{0.5}$	$55.5_{2.0}$	$89.2_{0.2}$	$69.6_{1.3}$	$93.8_{0.2}$	$87.8_{0.1}$	$71.1_{0.8}$	$71.5_{2.5}$	$59.7_{3.9}$	$92.5_{1.1}$
ReCoRD	$73.1_{0.9}$	$54.7_{1.3}$	$87.7_{0.7}$	$65.5_{0.9}$	$93.7_{0.1}$	**88.7**$_{0.3}$	$67.5_{1.3}$	**77.2**$_{2.3}$	$59.3_{1.2}$	$74.1_{5.2}$
CxC	$75.9_{0.4}$	$55.0_{0.2}$	**90.0**$_{0.0}$	**70.2**$_{0.1}$	**93.9**$_{0.2}$	$88.0_{0.4}$	$70.3_{0.5}$	$68.6_{2.5}$	$60.3_{3.9}$	$89.3_{2.4}$
SQuAD	$76.0_{0.7}$	$54.9_{1.2}$	$87.6_{0.1}$	$66.8_{0.3}$	**93.9**$_{0.5}$	**88.7**$_{0.7}$	$71.2_{0.4}$	$72.4_{0.5}$	$63.0_{1.6}$	$91.3_{1.3}$
DROP	$73.6_{1.3}$	$53.0_{1.0}$	$86.9_{0.9}$	$67.5_{1.2}$	$93.7_{0.2}$	$88.2_{0.3}$	$65.7_{3.1}$	$73.4_{2.0}$	$60.0_{3.6}$	$78.5_{8.6}$
SST-2	$73.3_{0.5}$	$52.3_{0.3}$	$87.9_{0.3}$	$63.8_{1.7}$	$93.8_{0.5}$	$85.6_{0.9}$	$66.9_{1.1}$	$68.6_{0.4}$	$57.0_{2.2}$	$92.9_{1.3}$
WinoGrande	$74.1_{0.8}$	$52.8_{1.6}$	$87.8_{0.3}$	$62.4_{2.5}$	$93.7_{0.1}$	$86.1_{0.5}$	$67.9_{1.3}$	$71.5_{2.5}$	$56.7_{1.2}$	$83.9_{0.8}$
HellaSWAG	$70.0_{2.6}$	$32.7_{23.6}$	$87.5_{0.2}$	$60.1_{3.9}$	$93.6_{0.0}$	$86.6_{1.4}$	$63.9_{5.4}$	$70.2_{2.1}$	$58.0_{2.2}$	$85.5_{2.6}$
MultiRC	$74.0_{0.5}$	$50.0_{4.6}$	$88.2_{0.2}$	$66.4_{0.5}$	$93.4_{0.1}$	$86.4_{1.3}$	$67.6_{1.0}$	$69.2_{4.1}$	$56.0_{4.1}$	$80.0_{8.6}$
CosmosQA	$73.4_{1.3}$	$52.1_{2.3}$	$87.7_{0.5}$	$65.9_{1.0}$	$93.6_{0.3}$	$87.9_{0.8}$	$68.7_{1.6}$	$69.6_{3.2}$	$62.3_{5.0}$	$83.9_{8.8}$
RACE	$73.6_{0.5}$	$52.5_{2.8}$	$87.5_{0.5}$	$63.1_{5.3}$	$93.4_{0.2}$	$86.5_{0.8}$	$66.5_{2.0}$	$68.9_{1.2}$	$57.3_{1.2}$	$84.8_{3.4}$

Table 6: Many tasks can benefit each other via prompt transfer. The orange-colored row shows the results of prompt tuning T5 Base on the target tasks from scratch (i.e., without any prompt transfer). Each cell in the other rows represents the target task performance when transferring the prompt from the associated source task (row) to the associated target task (column). Positive transfers are shown in green and the best results are highlighted in bold (green). Numbers in the subscript indicate the standard deviation across 3 random seeds.

Transfer	Increase (relative)
MNLI → CB	58.9
MNLI → COPA	29.1
ReCoRD → WSC	20.0
MNLI → RTE	19.2
ReCoRD → MRPC	18.7
SQuAD → MRPC	18.7
CxC → WiC	18.1
MNLI → BoolQ	17.0
MNLI → MRPC	16.5
QNLI → WiC	16.5
MNLI → WiC	16.2
CxC → STS-B	16.0
DROP → MRPC	15.1
SQuAD → COPA	14.5
QQP → MRPC	14.4
CxC → MRPC	13.7
C4 → MRPC	13.7
CosmosQA → MRPC	12.9
CosmosQA → COPA	12.9
QQP → COPA	12.2
QNLI → MRPC	12.2
QQP → WiC	11.8
MNLI → STS-B	11.8
SQuAD → BoolQ	11.1
QQP → STS-B	10.9
QQP → BoolQ	10.7
CxC → BoolQ	10.7
DROP → WiC	10.7
QQP → RTE	10.5
C4 → BoolQ	10.4

Table 7: Positive transfers with more than 10% relative error reduction on the target task. $s \rightarrow t$ denotes the transfer from source task s to target task t.

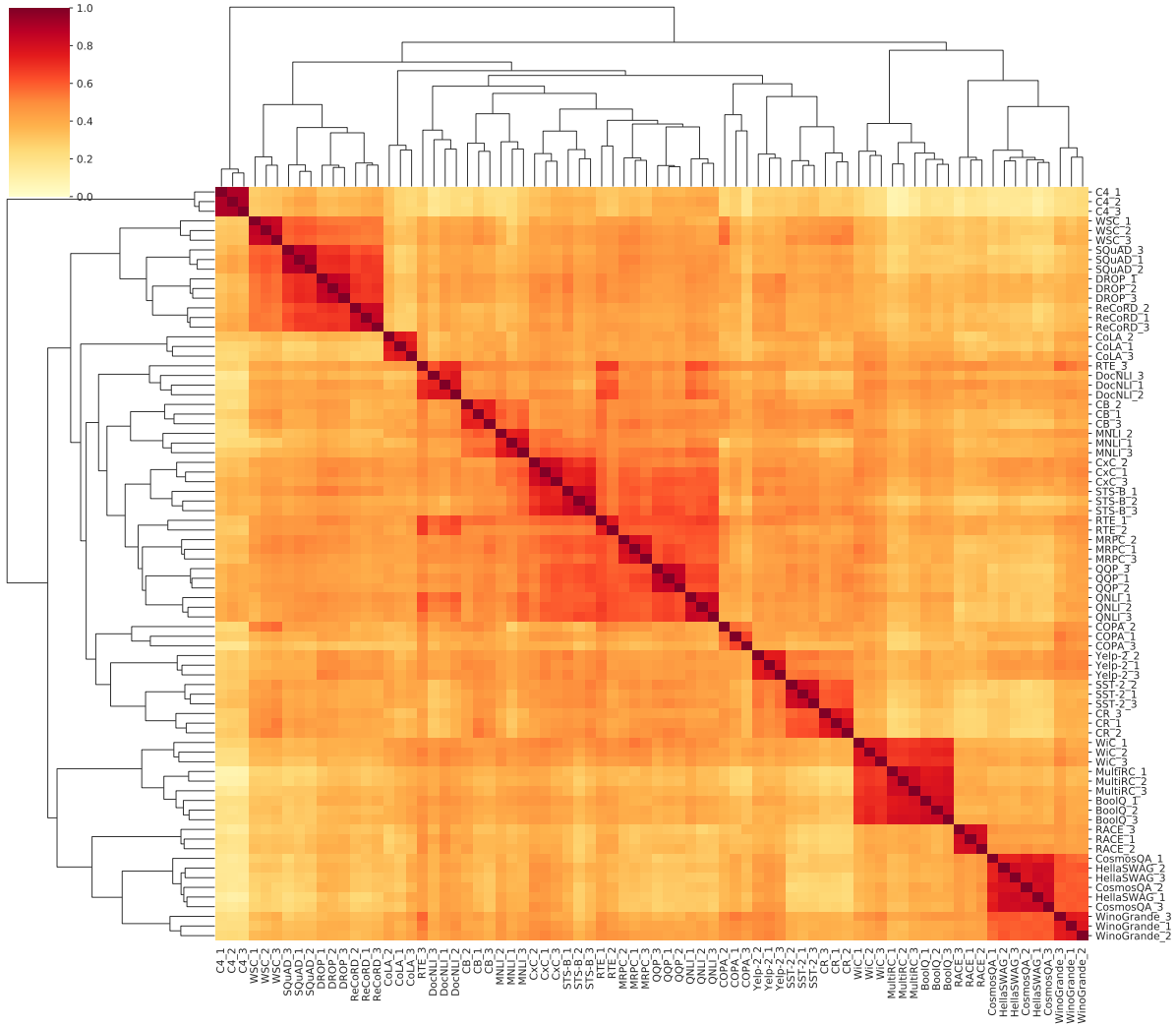

Figure 7: Our prompt-based task embeddings capture task relationships: similar tasks group together into clusters. Additionally, task embeddings that are derived from different prompts of the same task are linked together. t_1, t_2, t_3 correspond to three different prompt tuning runs on task t.

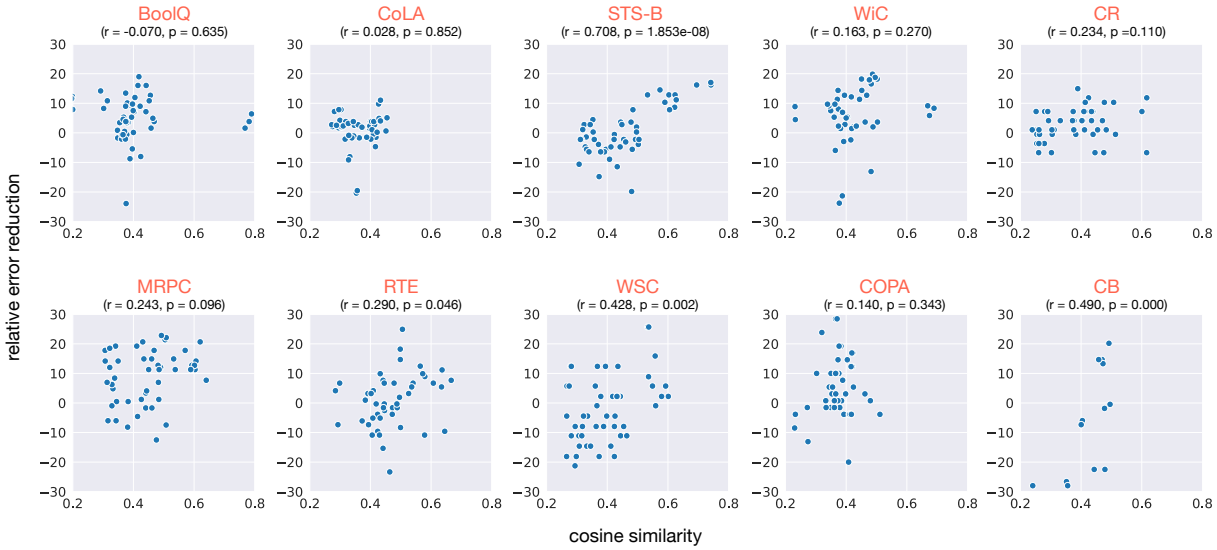

Figure 8: Correlation between task similarity and task transferability. Each point represents a source prompt. The x-axis shows the cosine similarity between the associated source and target task embeddings, averaged over three runs for the target task (orange title). The y-axis measures the relative error reduction on the target task achieved by each source prompt. We include the Pearson correlation coefficient (r) and p-value.