Exploring Low-dimensional Intrinsic Task Subspace via Prompt Tuning

Anonymous ACL submission

Abstract

Why can pre-trained language models (PLMs) learn universal representations and effectively adapt to broad NLP tasks differing a lot superficially? In this work, we empirically find evidence indicating that the adaptations of PLMs to various few-shot tasks can be reparameterized as optimizing only a few free parameters in a unified low-dimensional *intrinsic task subspace*, which may help us understand why PLMs could easily adapt to various NLP tasks with small-scale data. To find such a subspace and examine its universality, we propose an analysis pipeline called *intrinsic prompt tuning* (IPT). Specifically, we resort to the recent success of prompt tuning and decompose the soft prompts of multiple NLP tasks into the same low-dimensional nonlinear subspace, then we learn to adapt the PLM to unseen data or tasks by only tuning parameters in this subspace. In the experiments, we study diverse few-shot NLP tasks and surprisingly find that in a 5-dimensional subspace found with 100 tasks, by only tuning 5 free parameters, we can recover 87% and 65% of the full prompt tuning performance for 100 seen tasks (using different training data) and 20 unseen tasks, respectively, showing great generalization ability of the found intrinsic task subspace. Besides being an analysis tool, IPT could further bring practical benefits, such as improving the prompt tuning stability.

1 Introduction

Pre-trained language models (PLMs) have shown dominant performances on various natural language processing (NLP) tasks (Han et al., 2021; Min et al., 2021). After pre-training huge parameters on massive data, a PLM can effectively adapt to diverse downstream NLP tasks with small-scale data through full-parameter fine-tuning or parameter-efficient tuning methods (Lester et al., 2021; Houlsby et al., 2019). Nevertheless, the mechanisms behind such adaptations remain unclear. Why can PLMs learn universal representations through task-irrelevant pre-training objectives and easily adapt to diverse NLP tasks differing a lot? Towards answering this question, in this paper, we hypothesize that the adaptations of PLMs to various downstream tasks can be reparameterized as optimizing only a few free parameters in a unified low-dimensional parameter subspace, which we call *intrinsic task subspace* (Figure 1).

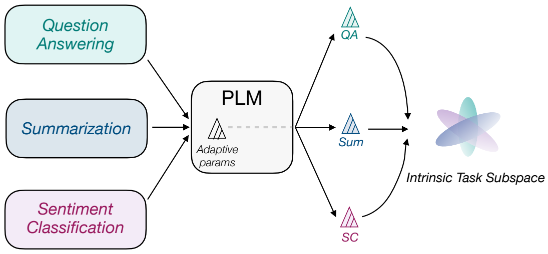


Figure 1: An illustration of a common low-dimensional intrinsic task subspace for diverse tasks. PLMs tune adaptive parameters to adapt to each task.

Specifically, during adaptation to a certain downstream task, PLMs optimize the tunable *adaptive parameters*. This is typically a high-dimensional optimization problem. For instance, in conventional fine-tuning, the adaptive parameters are all the PLM parameters, which may exceed hundreds of millions. However, Aghajanyan et al. (2021) show that the adaptation to a single task of a PLM can be reparameterized into only optimizing hundreds of free parameters in a low-dimensional subspace and then randomly projecting the tuned parameters back into the full parameter space. This motivates our hypothesis that adaptations to multiple tasks can be reparameterized into optimizations in **a unified** low-dimensional intrinsic task subspace. If this hypothesis holds, then (1) the existence of a common task reparameterization subspace explains the universality of PLMs and (2) the low dimensionality explains why the adaptations can be done with relatively small-scale data. From this perspective, the PLMs serve as general *compression frameworks*, which compress the learning complexity of various tasks from very high dimensionalities to low dimensionalities.

To find evidence for the hypothesis, we need to develop methods for finding the common intrinsic task subspaces of PLMs. Naturally, the subspace should contain adaptation solutions (i.e., tuned adaptive parameters) for various tasks, hence we can approximate the subspace by training a low-dimensional decomposition of the adaptive parameters using multiple tasks and then examine whether we can learn unseen tasks in the found subspace. However, training a decomposition for all the PLM parameters (the case of fine-tuning) is computationally unaffordable since the required parameters of the decomposition would be hundreds of times of PLMs. Fortunately, prompt tuning (PT) provides a parameter-efficient alternative, whose number of adaptive parameters (*soft prompts*), are only tens of thousands. PT can also achieve close performance to fine-tuning on both understanding (Lester et al., 2021) and generation (Li and Liang, 2021) tasks.

In experiments, we explore the common intrinsic subspace through PT under the few-shot learning setting, which ensures the data scales of various tasks are balanced. We name the analysis pipeline used in this paper as **I**ntrinsic **P**rompt **T**uning (**IPT**), which consists of two phases: multi-task subspace finding (MSF) and intrinsic subspace tuning (IST). During MSF, we first obtain trained soft prompts for multiple tasks and then learn an auto-encoder by first projecting them into the desired low-dimensional subspace and then reconstructing them with a back-projection. During IST, to adapt the PLM to unseen data and tasks, we only train the few free parameters in the low-dimensional subspace found by MSF through a fixed back-projection.

Surprisingly, we find that the intrinsic task subspace may not only exist but also is extremely low-dimensional. We study diverse few-shot NLP tasks and find that in a 5-dimensional subspace found by 100 tasks with MSF, we can recover 87% and 65% of the full PT performance with IST for 100 seen tasks (using different training data) and 20 unseen tasks, respectively. Furthermore, we analyze the effect of training task types, the number of training tasks, and training data scales for IPT. We also show that IPT and the intrinsic task subspace could bring some practical uses, such as analyzing task differences and improving training stability. We encourage future work to explore how to better find the intrinsic task subspace and develop techniques taking inspiration from low-dimensional reparameterizations of PLM adaptations.

2 Related Work

PLM, Fine-tuning and Prompt tuning. Since the success of BERT (Devlin et al., 2019), pre-trained language models bring a new paradigm to NLP, that is to pre-train a massive model as the universal backbone and then adapt the PLMs to specific downstream tasks. The mainstream way of downstream adaptation is fine-tuning, which adds task-specific classification heads and tunes all the PLM parameters with supervised data.

Recently, researchers found that promising results can be achieved by casting downstream tasks into the form of pre-training tasks and adding some *prompt* tokens into the input, including human-designed explainable prompts (Brown et al., 2020; Schick and Schütze, 2021a,b) and automatically searched prompts (Jiang et al., 2020; Shin et al., 2020; Gao et al., 2021). Following this line of study, the prompts are extended from real tokens to trainable embeddings, i.e., soft prompts (Hambardzumyan et al., 2021; Zhong et al., 2021; Qin and Eisner, 2021). Furthermore, some works (Lester et al., 2021; Li and Liang, 2021) demonstrate that only tuning soft prompts and keeping PLMs frozen can achieve excellent performance in various tasks, especially for large-scale PLMs. In this work, we try to understand these phenomena, i.e., why can PLMs learn universal abilities to adapt to various tasks with few data points and tunable parameters.

Intrinsic Dimensionality. *Intrinsic dimension* (ID) is the minimal number of variables needed to represent some data or approximate a function. Li et al. (2018) propose to measure the IDs of objective functions optimized by neural networks through randomly projecting all the trainable parameters into linear subspaces and finding the minimal dimensions that satisfactory solutions appear. Following this, Aghajanyan et al. (2021) show that the IDs of PLM adaptations (via fine-tuning) to many NLP tasks can be smaller than thousands and the pre-training implicitly lowers the IDs of downstream tasks, which motivates this work. Considering the existence of individual subspace for each task has been proved, here we aim to study whether

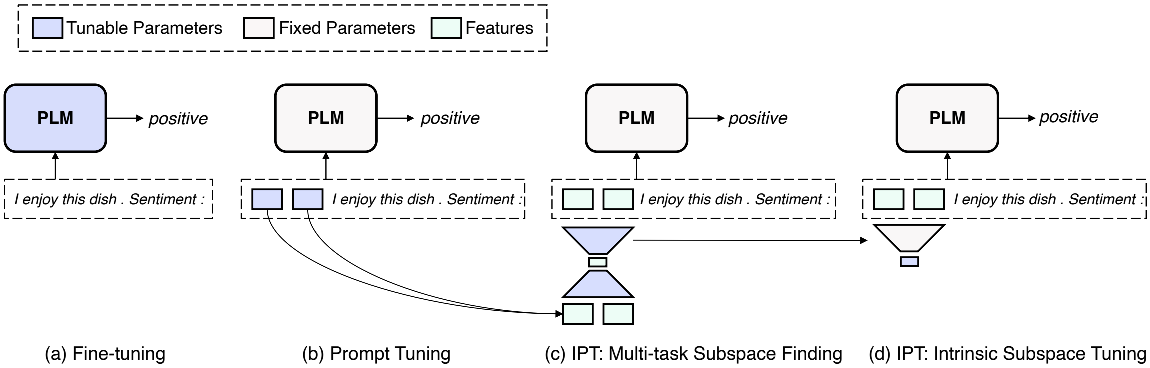


Figure 2: Illustrations of fine-tuning (a), prompt tuning (b) and two components of IPT (c,d). We discriminate tunable parameters, fixed parameters and intermediate features with different colors.

the subspace is universal. However, the random linear projections of previous methods inevitably introduce redundant task-irrelevant information and make the investigated subspace not compact for reparameterizing task adaptations. Therefore, we resort to stronger subspace-finding methods and use supervision from diverse tasks to train a nonlinear low-dimensional decomposition for the adaptive parameters.

Unifying Different NLP Tasks. Although various NLP tasks differ a lot on the surface, there has been long-standing attempts to unify different NLP tasks into the same form (Sun et al., 2021) and thus handle them with similar techniques, especially after the success of the prompting methods (Liu et al., 2021) to cast various tasks into the form of pre-training tasks of PLMs. The analyses in this paper may help us understand why can this be possible and explore how to better unify different tasks from the perspective of intrinsic task subspace.

3 Methodology

We first introduce essential preliminaries for both fine-tuning and prompt tuning in § 3.1, and then introduce our proposed analysis pipeline **I**ntrinsic **P**rompt **T**uning (**IPT**) in § 3.2, which consists of two stages: (1) Multi-task Subspace Finding (MSF) and (2) Intrinsic Subspace Tuning (IST). In Figure 2, we visualize the paradigms of fine-tuning, prompt tuning and our IPT.

3.1 Preliminaries

Assume we are given a series of NLP tasks $\{\mathcal{T}_1, \ldots, \mathcal{T}_{|\mathcal{T}|}\}$ partitioned into training tasks $\mathcal{T}_{\text{train}}$ and test tasks $\mathcal{T}_{\text{test}}$. Following Raffel et al. (2019), without loss of generality, we cast each task $\mathcal{T}_i$ into the unified conditional generation format. Given a training instance $(\mathcal{X}, \mathcal{Y})$ of $\mathcal{T}_i$, where both the input $\mathcal{X}$ and the target $\mathcal{Y}$ consist of a sequence of tokens, i.e., $\mathcal{X} = \{w_1, \ldots, w_{|\mathcal{X}|}\}$ and $\mathcal{Y} = \{y_1, \ldots, y_{|\mathcal{Y}|}\}$. Our goal is to learn a mapping function $\mathcal{F}_i : \mathcal{X} \rightarrow \mathcal{Y}$, and the de-facto way is to model $\mathcal{F}_i$ with a PLM $\mathcal{M}$, which first converts the input $\mathcal{X}$ into embeddings $\mathbf{E} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{X}|}\} \in \mathbb{R}^{|\mathcal{X}|\times d}$, where d denotes the input embedding dimension, then encodes $\mathbf{E}$ into hidden representations $\mathbf{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_{|\mathcal{X}|}\} \in \mathbb{R}^{|\mathcal{X}|\times d}$ and finally decodes $\mathcal{Y}$ conditioning on $\mathbf{H}$. The goal is to optimize the following objective:

$$\mathcal{L}_{\text{LM}} = -\frac{1}{|\mathcal{Y}|}\prod_{j=1}^{|\mathcal{Y}|} p(y_j|w_1, ..., w_{|\mathcal{X}|}, y_1, ..., y_{j-1}).$$

In traditional fine-tuning, all parameters of $\mathcal{M}$ ($\theta_{\mathcal{M}}$) are tuned during the optimization. Rather, prompt tuning (PT) prepends some task-specific embeddings (i.e., *soft prompts*) $\mathbf{P}_i = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ parameterized by θ_P before $\mathbf{E}$, and thus modify the input embeddings into $\mathbf{E}^* = \{\mathbf{p}_1, \ldots, \mathbf{p}_n; \mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{X}|}\} \in \mathbb{R}^{(n+|\mathcal{X}|)\times d}$. Then we keep $\theta_{\mathcal{M}}$ frozen and only tune θ_P to adapt $\mathcal{M}$ to $\mathcal{T}_i$ during PT. The training objective of PT is essentially the same as $\mathcal{L}_{\text{LM}}$ and denoted as $\mathcal{L}_{\text{LM}}(\mathbf{P}_i)$.

3.2 Intrinsic Prompt Tuning

To verify our hypothesis that the adaptations of PLMs to various downstream tasks can be reparameterized as optimization within a unified low-dimensional *intrinsic task subspace*, we propose a two-phase analysis pipeline IPT. The first phase MSF aims to find the intrinsic task subspace with multiple tasks' prompts, which are defined by an auto-encoder consisting of a projection function and a back-projection function. The second phase IST tunes a low-dimensional vector in the subspace and then recovers the vector to soft prompts through the back-projection function.

Multi-task Subspace Finding. We first conduct prompt tuning for each downstream task $\mathcal{T}_i$ and obtain the trained soft prompts $\mathbf{P}_i \in \mathbb{R}^{n\times d}$. During MSF, we try to find a satisfactory intrinsic task subspace of a low dimension d_I by learning a decomposition for the matrix $\mathbf{P}_i$. Inspired by text autoencoders (Bowman et al., 2016), the decomposition consists of a projection function $\mathbf{Proj}(\cdot)$ to project $\mathbf{P}_i$ into the d_I-dimensional subspace and a back-projection function $\mathbf{Proj}_b(\cdot)$ to project the d_I-dimensional vectors back into soft prompts of $\mathcal{T}_i$, and we optimize the reconstruction loss $\mathcal{L}^i_{\text{AE}}$:

$$\begin{aligned}\mathbf{P}^*_i &= \mathbf{Proj}_b(\mathbf{Proj}(\mathbf{P}_i)),\\ \mathcal{L}^i_{\text{AE}} &= ||\mathbf{P}^*_i - \mathbf{P}_i||^2_2,\end{aligned} \quad (255)$$

where $\mathbf{Proj}(\cdot)$ is implemented with a one-layer feed-forward network and $\mathbf{Proj}_b(\cdot)$ is parameterized by a two-layer nonlinear perceptron.

Moreover, finding the decomposition of a certain task's prompt $\mathbf{P}_i$, which is essentially a matrix, is somewhat trivial. Since the desired intrinsic task subspace should work for broad tasks, we introduce multi-task training and also take the task-oriented language modeling losses using the reconstructed soft prompts as objective functions. By jointly optimizing the reconstruction losses and the task-oriented losses, the subspace could gain the ability to reparameterize various task adaptations. The overall training objective of MSF is as follows:

$$\mathcal{L}^{\text{MSF}}_{\theta_{\text{proj}}} = \frac{1}{|\mathcal{T}_{\text{train}}|}\sum_{i=1}^{|\mathcal{T}_{\text{train}}|}(\mathcal{L}_{\text{LM}}(\mathbf{P}^*_i) + \alpha\mathcal{L}^i_{\text{AE}}), \quad (270)$$

where α denotes the hyper-parameter controlling the ratio between the two losses, and θ_{proj} denotes the parameters of both $\mathbf{Proj}$ and $\mathbf{Proj}_b$. During MSF, we only optimize θ_{proj} while keeping other parameters fixed. By introducing downstream task supervision and nonlinearity, we could find more irredundant and effective subspaces than the random linear subspaces (Aghajanyan et al., 2021).

Intrinsic Subspace Tuning. In this stage, we want to evaluate if the subspace found by MSF is generalizable to previously (1) unseen training data of $\mathcal{T}_{\text{train}}$ and (2) unseen tasks $\mathcal{T}_{\text{test}}$. And if the answer is yes, we can say that we successfully find the intrinsic task subspace reparameterizing the adaptations of PLMs to various tasks to some extent. Specifically, we only retain $\mathbf{Proj}_b$ learned during MSF and keep both $\mathbf{Proj}_b$ and $\mathcal{M}$ fixed. Then for each task $\mathcal{T}_i$, instead of conducting vanilla prompt tuning, we tune only d_I free parameters (θ_d) in the found subspace, which form an *intrinsic vector* $\mathbf{V}_i \in \mathbb{R}^{d_I}$, and project them into soft prompts with the fixed $\mathbf{Proj}_b$. The objective function for training a specific task $\mathcal{T}_i$ could be formulated as:

$$\mathcal{L}^{\text{IST}}_{\theta_d} = \mathcal{L}_{\text{LM}}(\mathbf{Proj}_b(\mathbf{V}_i)). \quad (294)$$

4 Experiment and Analysis

In this section, we first describe the experimental settings in § 4.1, including the tasks and corresponding datasets, evaluation metrics, evaluation pipeline and training details. Then we introduce the experimental results and analyses in § 4.2 and § 4.3.

4.1 Experimental Settings

Tasks and Datasets. To cover broad and diverse NLP tasks, we randomly choose 120 typical few-shot NLP tasks from *CrossFit Gym* (Ye et al., 2021). The few-shot setting ensures the data scales of tasks are balanced so that the subspace found by MSF will not be easily biased towards data-rich tasks.

For a brief introduction, *CrossFit Gym* consists of various types of few-shot NLP tasks, including text classification, question answering, conditional generation, etc. As mentioned in § 3.1, all tasks are processed into a unified sequence-to-sequence format following Raffel et al. (2019) and Khashabi et al. (2020) for ease of handling them with unified text-to-text PLMs. Each task $\mathcal{T}_i \in \mathcal{T}$ could be represented as a tuple of ($\mathcal{D}^i_{\text{train}}$, $\mathcal{D}^i_{\text{dev}}$, $\mathcal{D}^i_{\text{test}}$), and the sizes of $\mathcal{D}^i_{\text{train}}$ and $\mathcal{D}^i_{\text{dev}}$ are both set to K in the few-shot setting. For classification and regression tasks, $K = 16$, while for other categories of tasks, $K = 32$. We list task details in appendix F.

Evaluation Metrics. Since different tasks have distinct evaluation protocols (e.g., F1 score for discriminative tasks and BLEU for generative tasks typically), as suggested by Ye et al. (2021), we introduce average relative performance (E_{rel}) instead of absolute performance as the evaluation metric. The average absolute performance is also reported in appendix A.1 for reference. Specifically, let $\mathcal{T} = \{\mathcal{T}_1, ..., \mathcal{T}_{|\mathcal{T}|}\}$ be the evaluated tasks and $E_{\mathcal{T}_i}$ denotes the test score of $\mathcal{T}_i$ for IPT, $E_{\text{rel}} = \frac{1}{|\mathcal{T}|}\sum_{\mathcal{T}_i\in\mathcal{T}}\frac{E_{\mathcal{T}_i}}{E^*_{\mathcal{T}_i}}$, where $E^*_{\mathcal{T}_i}$ denotes the performance of either prompt tuning (in which we denote the final score as $E^{\text{PT}}_{\text{rel}}$) or fine-tuning ($E^{\text{FT}}_{\text{rel}}$).

Shorthand	$\mathcal{T}_{\text{train}}$	$\mathcal{T}_{\text{test}}$
random	100 random	20 random
non-cls	35 non-cls.	42 non-cls.($\mathcal{T}_{\text{test}}^{\text{in}}$) / 43 cls.($\mathcal{T}_{\text{test}}^{\text{out}}$)
cls	35 cls.	8 cls.($\mathcal{T}_{\text{test}}^{\text{in}}$) / 77 non-cls.($\mathcal{T}_{\text{test}}^{\text{out}}$)

Table 1: The overall 120 tasks $\mathcal{T}_{\text{all}}$ consist of 43 classification tasks (cls.) and 77 non-classification tasks (non-cls.). Three task splits are evaluated, including *random*, *non-cls* and *cls*, with details listed above, e.g., for *non-cls* partition, 35 non-cls. are chosen as $\mathcal{T}_{\text{train}}$ and 42 non-cls. / 43 cls. are chosen as $\mathcal{T}_{\text{test}}^{\text{in}}$ / $\mathcal{T}_{\text{test}}^{\text{out}}$, respectively.

Evaluation Pipeline. To properly evaluate the generalization ability achieved by IPT, we randomly split the overall task set $\mathcal{T}_{\text{all}}$ into training tasks $\mathcal{T}_{\text{train}}$ and test tasks $\mathcal{T}_{\text{test}}$. We adopt three task splits as introduced in Table 1 to investigate the influence of task types. We first conduct prompt tuning on all tasks and obtain the trained soft prompts.

During MSF, we train $\mathbf{Proj}$ and $\mathbf{Proj}_b$ on $\mathcal{T}_{\text{train}}$ only, and evaluate the reconstructed prompts on $\mathcal{T}_{\text{train}}$ (denoted as $\mathcal{T}_{\text{train}}$(MSF)) to see how much performance we will lose in the process of reconstructing prompts from d_I-dimensional subspace, which will provide an empirical upper bound for the generalization to unseen data and tasks in our setting. We also directly reconstruct the soft prompts of $\mathcal{T}_{\text{test}}$ with the learned auto-encoder and test their performance ($\mathcal{T}_{\text{test}}$(MSF)) to see the auto-encoder's reconstruction ability for unseen soft prompts.

For IST, we first carry out experiments on $\mathcal{T}_{\text{train}}$ using exactly the same $\mathcal{D}_{\text{train}}^{i}$ / $\mathcal{D}_{\text{dev}}^{i}$ utilized in MSF training and get a result $\mathcal{T}_{\text{train}}^{\text{same}}$(IST). After that, we evaluate the generalization ability of IPT to see whether adaptations to various tasks are substantially reparameterized into the found subspace with two generalization challenges: (1) *unseen-data challenge* and (2) *unseen-task challenge*.

• For the *unseen-data challenge*, we sample different training and validation data for $\mathcal{T}_{\text{train}}$ while keeping test data the same. Then we conduct IST with the new data and test its performance on $\mathcal{T}_{\text{train}}$, which is denoted as $\mathcal{T}_{\text{train}}^{\text{diff}}$(IST). This challenge is to test whether the learned subspace can also reparameterize optimization on unseen data, which naturally has different optimization trajectories.

• For the *unseen-task challenge*, we evaluate the soft prompts obtained by IST on $\mathcal{T}_{\text{test}}$, which are tasks unseen during MSF, to see how well can optimization in the found subspace recover PLM adaptations of unseen tasks, which will provide evidence for our hypothesis that the reparameterization subspaces for different task adaptations are not orthogonal. In the *random* split, the results are denoted as $\mathcal{T}_{\text{test}}$(IST). In the *non-cls* and *cls* splits, we have two test sets with different task types and the results are denoted as $\mathcal{T}_{\text{test}}^{\text{in}}$(IST) and $\mathcal{T}_{\text{test}}^{\text{out}}$(IST).

Training Details. Since all tasks are unified into the same sequence-to-sequence format, we use BART$_{\texttt{BASE}}$ (Lewis et al., 2020) for the experiments in the main paper and also test BART$_{\texttt{LARGE}}$ in appendix A.3. For the prompt tuning / fine-tuning baseline, we perform grid search on the combination of a series of learning rates and batch sizes and choose the best checkpoint using $\mathcal{D}_{\text{dev}}$. We set the number of soft prompts to be 100 for all tasks and randomly initialize them. For IPT, we examine the dimension d_I of $\{3, 5, 10, 50, 100\}$. Note that for fine-tuning / prompt tuning, 139M / $76,800$ parameters are tuned, while IPT only tunes d_I free parameters. More details are left in appendix D.

4.2 Main Results

Based on the experimental results shown in Figure 3, we study the following questions:

Q1. Do PLMs really reparameterize various task adaptations into a low-dimensional task subspace in the few-shot setting? From the results in Figure 3 (a), we observe that: (1) for the *unseen-data challenge* ($\mathcal{T}_{\text{train}}^{\text{diff}}$(IST)), when $d_I \geq 5$, IST on unseen i.i.d. data could recover more than 80% of the full prompt tuning performance of the 100 training tasks; (2) for the *unseen-task challenge* ($\mathcal{T}_{\text{test}}$(IST)), we can also achieve about 60% performances by only tuning $5 \sim 100$ parameters. From these results, we can say that the low-dimensional reparameterizations in the subspaces found by MSF successfully recover the PLM adapations of $\mathcal{T}_{\text{train}}$ and can also generalize to unseen tasks to some extent, thus non-trivial performances can be achieved by only tuning a few free parameters in these subspaces. This provides evidence for our hypothesis that PLMs reparameterize various task adaptations into the same low-dimensional subspace, or at least the low-dimensional reparameterization subspaces for various task adaptations (Aghajanyan et al., 2021) should have a substantial intersection, otherwise the subspaces found by $\mathcal{T}_{\text{train}}$ will be almost impossible to also work for $\mathcal{T}_{\text{test}}$.

Q2. What limits IPT? Although positive evidence is observed, the effectiveness of IPT is still limited considering only about 60% performances

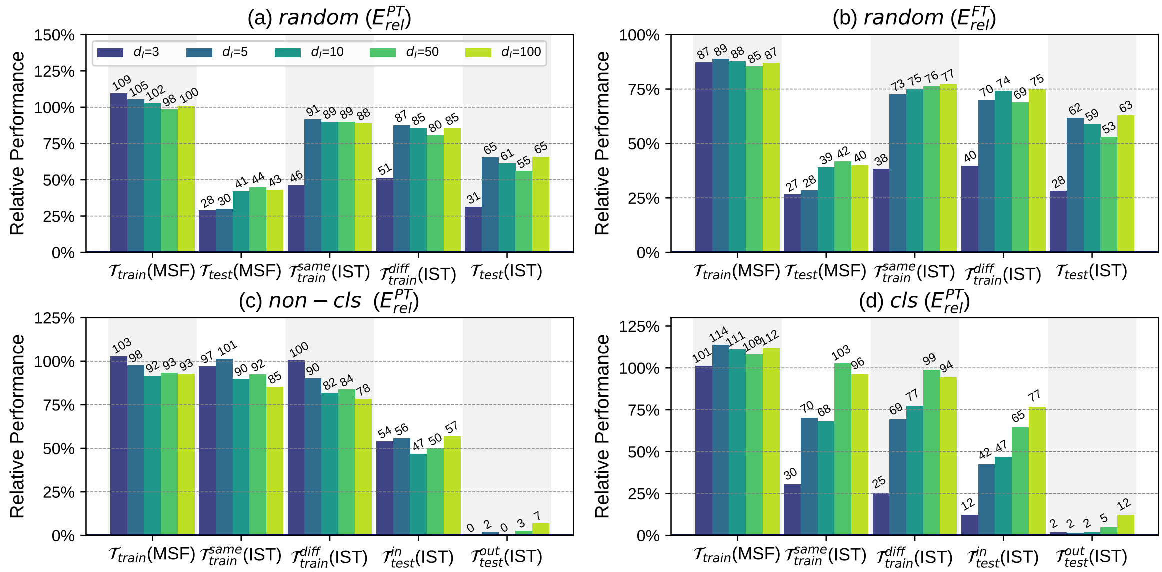


Figure 3: Relative performance of IPT at different dimension d_I on three task splits (*random*, *non-cls* and *cls*). We report the relative performance of IPT comparing with both prompt tuning ($E_{\text{rel}}^{\text{PT}}$) and fine-tuning ($E_{\text{rel}}^{\text{FT}}$).

can be recovered for unseen tasks. From the results in Figure 3 (a) and (b), we discuss what factors may limit the effectiveness of IPT and provide insights for improving the analysis pipeline.

(1) **Reconstruction ability of the auto-encoder.** The performance on $\mathcal{T}_{\text{train}}$ when we directly reconstruct soft prompts using the auto-encoder of MSF ($\mathcal{T}_{\text{train}}$(MSF)) are even better than vanilla prompt tuning (PT), which demonstrates that MSF can improve PT by enforcing multi-task skill sharing within the extremely low dimensions. In addition, from the comparisons between $\mathcal{T}_{\text{train}}$(MSF) and $\mathcal{T}_{\text{test}}$(MSF), we can see that directly reconstructing soft prompts of unseen tasks performs poorly. It indicates that the reconstruction ability of the auto-encoders trained in MSF cannot generalize well to unseen soft prompts, which will limit IPT to some extent. This may come from the MSF training methods and the limited representation ability of the networks used to parameterize $\mathbf{Proj}(\cdot)$ and $\mathbf{Proj}_b(\cdot)$. Nevertheless, IST could find much better solutions than MSF reconstructed prompts with task-specific supervisions on $\mathcal{T}_{\text{test}}$.

(2) **Optimization in IST.** The consistently higher performance of $\mathcal{T}_{\text{train}}$(MSF) over $\mathcal{T}_{\text{train}}^{\text{same}}$(IST) and $\mathcal{T}_{\text{train}}^{\text{diff}}$(IST) demonstrates that there exists good enough solutions for $\mathcal{T}_{\text{train}}$ in the found subspaces. However, even using exactly the same training data, IST cannot find these good solutions (the gap between $\mathcal{T}_{\text{train}}$(MSF) and $\mathcal{T}_{\text{train}}^{\text{same}}$(IST)), which shows that the adopted optimization algorithm limits the performance of IST to some extent.

(3) **Adaptive parameters.** Comparing the results in Figure 3 (a) and (b), we observe that the recovered relative performance of fine-tuning ($E_{\text{rel}}^{\text{FT}}$) is always poorer than that of PT ($E_{\text{rel}}^{\text{PT}}$). This is because PT is slightly inferior than fine-tuning under the few-shot setting, and the performance of IPT is bounded by PT since MSF is based on decomposing soft prompts. Ideally, $E_{\text{rel}}^{\text{FT}}$ could be improved by designing better PT algorithms or selecting more appropriate adaptive parameters.

Q3. How is the influence of task types? Following Ye et al. (2021), we divide the studied tasks into *cls* (classification), which are discriminative tasks and *non-cls* (non-classification), which tend to be generative tasks. From the results in Figure 3 (c)-(d), we find that: (1) there exists a huge generalization gap between *cls* tasks and *non-cls* tasks. When using only one kind of tasks during MSF, the found subspaces work well for the same kind of tasks ($\mathcal{T}_{\text{test}}^{\text{in}}$(IST)) but generalize poorly to the other kind of tasks ($\mathcal{T}_{\text{test}}^{\text{out}}$(IST)). This shows that the found subspace is severely biased by the training task types. (2) When increasing d_I, *cls* performance (Figure 3 (d)) tends to increase, but *non-cls* performance (Figure 3 (c)) tends to decrease. The opposite trends of these two types of tasks make the IPT performance on the *random* split exhibit a constant trend when $d_I \geq 5$. Intuitively, the ideal common reparameterization subspace for multiple task adaptations has an optimal dimension $\hat{d}$. When

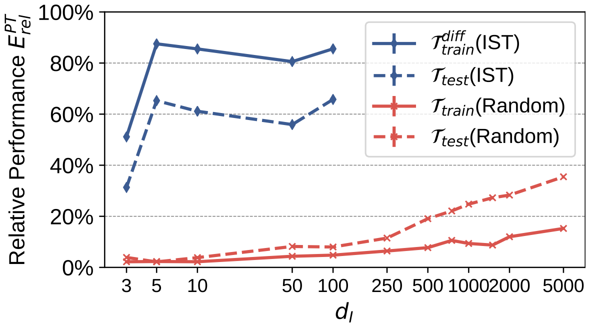


Figure 4: Comparisons between IPT and randomly generated subspaces on the *random* task split.

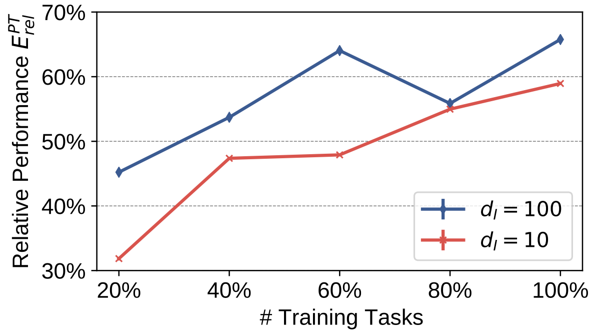


Figure 5: Impacts of the number of training tasks.

$d_I < \hat{d}$, the d_I-dimensional subspace is not strong enough to reparameterize these task adaptations and thus increasing d_I leads to better IPT performance. When $d_I > \hat{d}$, since MSF must decompose the soft prompts into a d_I-dimensional subspace, MSF is likely to put some redundant and confounding information into the found subspace and thus results in the decrease of IPT performance[1]. Hence this indicates that, although counter-intuitively, the $\hat{d}$ for *non-cls* tasks is far smaller than *cls* tasks. We hypothesize this may come from the few-shot setup and will explore it in the future.

4.3 Analyses and Properties

Comparison with Random Subspace. Previous works (Li et al., 2018; Aghajanyan et al., 2021) adopt randomly generated subspaces and avoid computation in subspace finding. While in this work, we introduce supervisions from diverse tasks to find the universal low-dimensional intrinsic task subspaces. To verify the effectiveness and necessity of task-specific supervisions in MSF, we compare IPT with conducting IST in randomly generated subspaces, which are defined by randomly initialized auto-encoders of the same architecture with the ones used in MSF. We compare them under the *random* task split. For IPT, we report the *unseen-data* ($\mathcal{T}^{\text{diff}}_{\text{train}}$(IST)) and *unseen-task* ($\mathcal{T}_{\text{test}}$(IST)) performance. For random subspaces, we also report their performance on $\mathcal{T}_{\text{train}}$ (denoted as $\mathcal{T}_{\text{train}}$(Random)) and $\mathcal{T}_{\text{test}}$ ($\mathcal{T}_{\text{test}}$(Random)), respectively. The results are shown in Figure 4, from which we can see that IPT could perform much better than random subspaces using much fewer dimensions, which indicates the effectiveness of MSF to exclude redundant task-irrelevant information and find compact reparameterization subspaces.

[1]We also observe non-increasing trends for the performance of *cls* task split when d_I is enlarged above 500.

Impacts of the Number of Training Tasks. During MSF, the auto-encoder is optimized to reparameterize the adaptive parameters of various training tasks. Ideally, the coverage of $\mathcal{T}_{\text{train}}$ would significantly impact the generalization ability of IPT on unseen tasks $\mathcal{T}_{\text{test}}$. To demonstrate this, we randomly sample $\{20\%, 40\%, 60\%, 80\%\}$ tasks from $\mathcal{T}_{\text{train}}$ of the *random* task split for training the auto-encoder, then evaluate IPT ($d_I = \{10, 100\}$) on original $\mathcal{T}_{\text{test}}$ with the *unseen-task* challenge. From the results visualized in Figure 5, we observe that with the number of training tasks growing, the generalization ability of the found intrinsic task subspace generally improves. This reflects that increasing the coverage and diversity of seen tasks could help IPT find more universal subspaces.

Impacts of the Data Scale. Although we adopt the few-shot setup to control the influence of data amount in this paper, it is also interesting to investigate whether IPT's ability could be further improved with more training data. Here we take an initial trial using the task split *cls* by doubling and quadrupling the number of data shots K (from 16 to 32 and 64), and investigate the performance of MSF ($\mathcal{T}_{\text{train}}$(MSF)) as well as IST under the *unseen-data* ($\mathcal{T}^{\text{diff}}_{\text{train}}$(IST)) and *unseen-task* ($\mathcal{T}^{\text{in}}_{\text{test}}$(IST)) challenges. Note that with different number of data points, the prompt tuning performance (denominator of $E^{\text{PT}}_{\text{rel}}$) is also different. The results are shown in Figure 6, from which we observe that when the data scale grows, the performance of IPT on *unseen data* and *unseen task* challenges generally become better, which shows the subspaces found with more data are more universal. Hence we believe it is interesting to explore in future how strong the performance of IPT on data-rich scenarios will be.

Visualization of the Found Intrinsic Subspace. We visualize the intrinsic vectors $\mathbf{V}_i$ (vectors con-

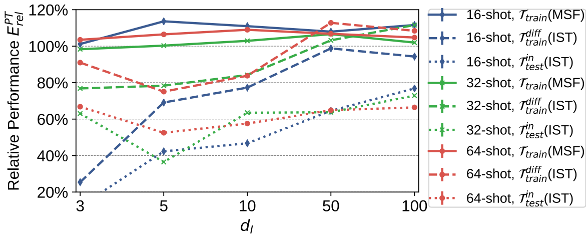


Figure 6: Impacts of the data scale.

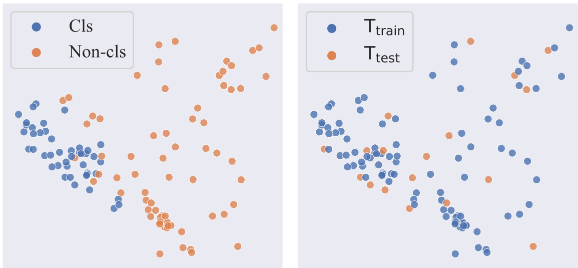


Figure 7: PCA plots of the intrinsic vectors learned during IST. We label points with different colors to represent its corresponding categories. Specifically, we show the clusters of (1) classification and non-classification tasks (left) and (2) $\mathcal{T}_{\text{train}}$ and $\mathcal{T}_{\text{test}}$ (right). Without loss of generality, we choose the task split of *random* and $d_I = 100$.

sisting of the free parameters learned during IST in the found subspace) using PCA in Figure 7, from which we observe that: (1) there exists a clear dividing line between the clusters of classification tasks and non-classification tasks, indicating that they are highly distinct, which also explains why subspaces learned on either cluster generalize poorly to the other cluster; (2) the points of unseen tasks $\mathcal{T}_{\text{test}}$ are mixed with those of $\mathcal{T}_{\text{train}}$, which demonstrates that the found subspaces universally reparameterize various tasks so that IPT can generalize to unseen tasks. We also visualize the clusters of fine-grained categories of QA and text classification tasks in appendix B. We argue that the learned intrinsic vectors could be viewed as low-dimensional task representations, helping us analyze the similarity and differences for various NLP tasks.

Improving Prompt Tuning Stability with IPT. In Table 2, we show the mean standard deviations (`std`) of test scores for 120 few-shot tasks over 10 runs comparing IPT ($d_I = 10$), fine-tuning and prompt tuning (PT). We observe that PT is the most unstable strategy with the highest `std`, while fine-tuning is far more stable. The instability of PT may influence its practical use. Intuitively, IPT only tunes a few free parameters, which will conduce

Method	$\mathcal{T}_{\text{train}}$	$\mathcal{T}_{\text{test}}$	$\mathcal{T}_{\text{all}}$
Fine-tuning	2.16	2.40	2.20
Prompt Tuning	3.06	4.19	3.25
IPT	**1.12**	**0.73**	**1.06**

Table 2: Standard deviations (`std`) of test scores over multiple runs. d_I of IPT is chosen to be 10.

to improving the stability, and IPT surely becomes the most stable method in Table 2. We further show in appendix A.4 that IPT and vanilla PT could be combined in a two-stage manner to improve both stability and performance.

5 Conclusion and Future Work

Could few-shot NLP tasks be reparameterized into a unified subspace? We study the hypothesis that PLM adaptations to various tasks can be reparameterized as optimizations within a **unified** low-dimensional *intrinsic task subspace*. We develop an analysis tool IPT. It first finds a subspace by jointly decomposing the adaptive parameters of multiple tasks and then tunes parameters within the subspace for unseen data and tasks. Experiments show that the found subspace contains sub-optimal but non-trivial solutions for PLM adaptations, which are strong evidence for our hypothesis.

However, we only investigate one PLM adaptation method, i.e., prompt tuning in this paper, and the achieved performance of IPT is still far from perfect. Although it may come from the inadequacy of current subspace-finding methods and optimization algorithms as mentioned in our analyses, based on current results, we cannot directly conclude that the hypothesis is true. Nevertheless, at least we have found promising empirical results showing that the low-dimensional reparameterization subspaces of various tasks have a substantial **intersection**, which MSF is designed to find.

What's next? In future, we will explore (1) how to improve IPT to find stronger evidence for our hypothesis, (2) whether the conclusions hold for other PLM adaptation methods like the adapter (Houlsby et al., 2019) and (3) whether the **union** of reparameterization subspaces for various tasks is also low-dimensional. We also encourage further explorations based on our hypothesis, such as (1) understanding the scaling law of PLMs, (2) how to utilize and manipulate intrinsic vectors, and (3) how to better tune PLMs in the intrinsic task subspaces. We leave the detailed discussions in appendix C.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. 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 7319–7328, Online. Association for Computational Linguistics.

Tiago A. Almeida, José María G. Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: New collection and results. In *Proceedings of the 11th ACM Symposium on Document Engineering*, DocEng '11, page 259–262, New York, NY, USA. Association for Computing Machinery.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. 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 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Victor Zhong an. 2017. Seq2sql: Generating structured queries from natural language usin. *ArXiv preprint*, abs/1709.00103.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*, volume 6, pages 6–4. Venice.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *Proceedings of Text Analysis Conference*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Michael Boratko, Xiang Li, Tim O'Gorman, Rajarshi Das, Dan Le, and Andrew McCallum. 2020. ProtoQA: A question answering dataset for prototypical common-sense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1122–1136, Online. Association for Computational Linguistics.

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. 2018. Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 732–737, Brussels, Belgium. Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Michael Chen, Mike D'Arcy, Alisa Liu, Jared Fernandez, and Doug Downey. 2019. CODAH: An adversarially-authored question answering dataset for common sense. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 63–69, Minneapolis, USA. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv preprint*, abs/1803.05457.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential 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)*, pages 5925–5932, Hong Kong, China. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

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.

T. Diggelmann, Jordan L. Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims. *ArXiv preprint*, abs/2012.00614.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Matthew Dunn, Levent Sagun, Mike Higgins, V. U. Güney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *ArXiv preprint*, abs/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*, pages 421–426, Tokyo, Japan. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *Computer Speech & Language*, 59:123–156.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

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*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Manaal Faruqui and Dipanjan Das. 2018. Identifying well-formed natural language questions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 798–803, Brussels, Belgium. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of* *the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

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*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885–892. Text Mining and Natural Language Processing in Pharmacogenomics.

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 (Volume 1: Long Papers)*, pages 4921–4933, Online. Association for Computational Linguistics.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. 2021. Pre-trained models: Past, present and future. *ArXiv preprint*, abs/2106.07139.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*.

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)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. 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 318–323, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *ArXiv preprint*, abs/2001.08361.

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, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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*, 7:453–466.

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*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, D. Kontokostas, Pablo N. Mendes, Sebastian Hellmann, M. Morsey, Patrick van Kleef, S. Auer, and C. Bizer. 2015. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195.

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

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'12, page 552–561. AAAI Press.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. 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.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. 2018. Measuring the intrinsic dimension of objective landscapes. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020a. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6862–6868, Online. Association for Computational Linguistics.

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

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 58–62, Hong Kong, China. Association for Computational Linguistics.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of* the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Irene Manotas, Ngoc Phuoc An Vo, and Vadim Sheinin. 2020. LiMiT: The literal motion in text dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 991–1000, Online. Association for Computational Linguistics.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *ArXiv preprint*, abs/2012.10289.

Clara H. McCreery, Namit Katariya, Anitha Kannan, Manish Chablani, and Xavier Amatriain. 2020. Effective transfer learning for identifying similar questions: Matching user questions to COVID-19 faqs. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3458–3465. ACM.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv preprint arXiv:2111.01243*.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. Ethos: an online hate speech detection dataset. *ArXiv preprint*, abs/2006.08328.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada. Association for Computational Linguistics.

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*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

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*, pages 4885–4901, Online. Association for Computational Linguistics.

Achraf Othman and Mohamed Jemni. 2012. English-asl gloss parallel corpus 2012: Aslg-pc12.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Dimitris Pappas, Petros Stavropoulos, Ion Androutsopoulos, and Ryan McDonald. 2020. BioMRC: A dataset for biomedical machine reading comprehension. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 140–149, Online. Association for Computational Linguistics.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Quan Hung Tran, and Thien Huu Nguyen. 2020. What does this acronym mean? introducing a new dataset for acronym identification and disambiguation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3285–3301, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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*, pages 5203–5212, Online. Association for Computational Linguistics.

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

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The Winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789, Jeju Island, Korea. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

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

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to ai complete question answering: A set of prerequisite real tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8722–8731.

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

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*, 34:8732–8740.

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)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. 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.

Timo Schick and Hinrich Schütze. 2021b. 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*, pages 2339–2352, Online. 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.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. 2019. Mining discourse markers for unsupervised sentence representation learning. 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 3477–3486, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Tianxiang Sun, Xiangyang Liu, Xipeng Qiu, and Xuanjing Huang. 2021. Paradigm shift in natural language processing. *ArXiv preprint*, abs/2109.12575.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2019a. Quarel: A dataset and models for answering questions about qualitative relationships. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7063–7071.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019b. QuaRTz: An open-domain dataset of qualitative relationship questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5941–5946, Hong Kong, China. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

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

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. *ArXiv preprint*, abs/2109.01652.

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

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

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

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7163–7189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

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*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Hao Zhang, Jae Ro, and Richard Sproat. 2020. Semi-supervised URL segmentation with recurrent neural networks pre-trained on knowledge graph entities. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4667–4675, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sheng Zhang, X. Liu, J. Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record:

Bridging the gap between human and machine commonsense reading comprehension. *ArXiv preprint*, abs/1810.12885.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China. Association for Computational Linguistics.

Appendices

A Additional Experiments

A.1 Absolute Performance

In the experiments, we mainly report the relative performance (E_{rel}). For reference, we also report the average absolute performance (E_{abs}) in this section. Let $E_{\mathcal{T}_i}$ denote the test score of $\mathcal{T}_i$ for IPT, $E_{\text{abs}} = \frac{1}{|\mathcal{T}|} \sum_{\mathcal{T}_i \in \mathcal{T}} E_{\mathcal{T}_i}$. The E_{abs} of BART$_{\texttt{BASE}}$ for prompt tuning and fine-tuning are shown in Table 3, and the E_{abs} of IPT on three task splits are shown in Table 4, Table 5 and Table 6, respectively.

A.2 Relative Performance to Fine-tuning

In the experiments, we report the relative performance to **prompt tuning** as the main evaluation metric except in Figure 3 (b), which reports the relative performance to **fine-tuning** on the *random* split for analyses. In this section, we additional report the $E_{\text{rel}}^{\text{FT}}$ on *non-cls* and *cls* splits in Figure 8 for reference, where we can see the general conclusions are consistent with our analyses in § 4.2.

A.3 BART$_{\texttt{LARGE}}$ Performance

All the experiments in § 4 are conducted with BART$_{\texttt{BASE}}$ model (Lewis et al., 2020), which is also the main evaluated model of our adopted evaluation platform *CrossFit* (Ye et al., 2021). To see whether the conclusions will also hold for larger models, we take a prior trial by conducting experiments on BART$_{\texttt{LARGE}}$. As the results shown in Figure 9 suggest, the overall conclusions are consistent with those of BART$_{\texttt{BASE}}$ that non-trivial performance can be recovered in the found subspaces. However, the performance is obviously worse than the cases of BART$_{\texttt{BASE}}$ when d_I is extremely low ($3 \sim 10$), especially on the *cls* split. This phenomenon may come from the difficulty of finding intrinsic task subspaces for larger PLMs, which is worthwhile to explore in the future.

A.4 Combining IPT and Vanilla Prompt Tuning

To make the stability advantage brought by IPT practical, we propose to use the solutions found by IPT as the initialization for the vanilla prompt tuning. Specifically, we continue the experiments of split *random* on $\mathcal{T}_{\text{test}}$ choosing $d_I = 10$ and initialize the soft prompts by back-projecting the found solutions in the subspace during IST. Other details are kept the same as the prompt tuning (PT) baseline. We observe that the standard variance achieved in this way is significantly lower than the vanilla PT (1.65 v.s. 4.19) while we can achieve 103.4% of $E_{\text{rel}}^{\text{PT}}$, i.e., the performance could also be improved from 59% (IST). This indicates that both IPT and vanilla PT could be further combined in a two-stage manner to improve both the training stability and performance. This experiment also demonstrates that although our IPT pipeline mainly works as an analytical framework in this paper, it can also bring practical benefits. We will explore more practical uses of IPT in the future.

Split	Prompt Tuning $\mathcal{T}_{\text{train}}$	Prompt Tuning $\mathcal{T}_{\text{test}}^{\text{in}}$ / $\mathcal{T}_{\text{test}}^{\text{out}}$	Fine-tuning $\mathcal{T}_{\text{train}}$	Fine-tuning $\mathcal{T}_{\text{test}}^{\text{in}}$ / $\mathcal{T}_{\text{test}}^{\text{out}}$
random	32.6	40.1 ($\mathcal{T}_{\text{test}}$)	35.2	40.7 ($\mathcal{T}_{\text{test}}$)
non-cls	23.0	28.0 / 49.0	24.4	29.6 / 52.2
cls	48.6	50.9 / 25.7	52.5	51.1 / 27.2

Table 3: Average absolute performance for prompt tuning / fine-tuning on the three task splits we adopted.

Dim (d_I)	3	5	10	50	100
Multi-task Projection Learning					
$\mathcal{T}_{\text{train}}$	29.1	31.8	32.2	32.0	**32.6**
$\mathcal{T}_{\text{test}}$	8.5	10.0	15.0	**16.7**	16.4
Single-task Intrinsic Subspace Tuning					
$\mathcal{T}_{\text{train}}^{\text{same}}$	13.2	25.6	27.8	28.8	**29.6**
$\mathcal{T}_{\text{train}}^{\text{diff}}$	13.0	24.9	27.4	26.7	**28.4**
$\mathcal{T}_{\text{test}}$	9.3	**26.5**	24.7	23.1	25.8

Table 4: Average absolute performance on the *random* task split.

Dim (d_I)	3	5	10	50	100
Multi-task Projection Learning					
$\mathcal{T}_{\text{train}}$	**23.3**	23.1	21.9	22.7	22.2
Single-task Intrinsic Subspace Tuning					
$\mathcal{T}_{\text{train}}^{\text{same}}$	22.1	**23.3**	21.4	20.4	19.5
$\mathcal{T}_{\text{train}}^{\text{diff}}$	**22.0**	20.5	17.4	19.6	19.7
$\mathcal{T}_{\text{test}}^{\text{in}}$	16.7	16.4	14.8	17.0	**19.5**
$\mathcal{T}_{\text{test}}^{\text{out}}$	0.0	1.0	0.8	1.4	**3.9**

Table 5: Average absolute performance on the *non-cls* task split.

B Additional Visualization

We visualize the intrinsic vectors of fine-grained categories of QA and text classification tasks using PCA in Figure 10. We observe that the same category points exhibit a compact cluster. This further shows that the learned intrinsic vectors could serve as task representations and help us analyze the similarity and differences for diverse NLP tasks.

Dim (d_I)	3	5	10	50	100
Multi-task Projection Learning					
$\mathcal{T}_{\text{train}}$	46.0	**50.0**	48.0	49.5	48.7
Single-task Intrinsic Subspace Tuning					
$\mathcal{T}^{\text{same}}_{\text{train}}$	12.2	32.0	30.3	**48.5**	47.2
$\mathcal{T}^{\text{diff}}_{\text{train}}$	10.5	33.0	31.9	**46.9**	44.4
$\mathcal{T}^{\text{in}}_{\text{test}}$	7.8	21.0	24.5	32.7	**38.1**
$\mathcal{T}^{\text{out}}_{\text{test}}$	0.6	0.7	1.0	2.1	**4.2**

Table 6: Average absolute performance on the *cls* task split.

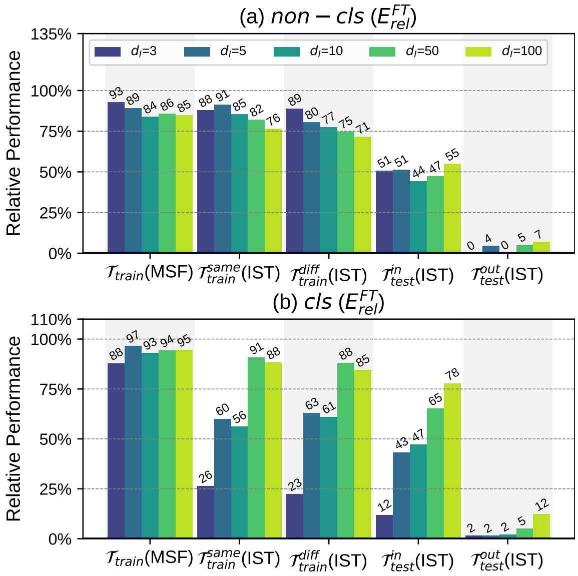


Figure 8: Relative performance of IPT at different dimension d_I on *non-cls* and *cls* splits, comparing with fine-tuning.

C Additional Discussion

Relation to the scaling law. Recently, researchers have found that larger PLMs tend to be more sample-efficient (Kaplan et al., 2020), parameter-efficient (Lester et al., 2021) and cross-task generalizable (Wei et al., 2021). Our hypothesis may help us understand this phenomenon: the adaptations of larger PLMs can be better reparameterized into a unified subspace so that the cross-task generalization will be easier, and larger PLMs have lower reparameterization dimensions (Aghajanyan et al., 2021), hence they should need fewer data and tunable parameters. This also implies that the characteristics of intrinsic task subspaces may be used to examine how well a PLM is trained.

Utilize and manipulate intrinsic vectors. The intrinsic vectors obtained during IST depict the adaptations to different tasks and it is worthwhile to explore whether we can (1) utilize them to find the relations among different tasks, and (2) manipulate these vectors to achieve some interesting cross-task generalization results.

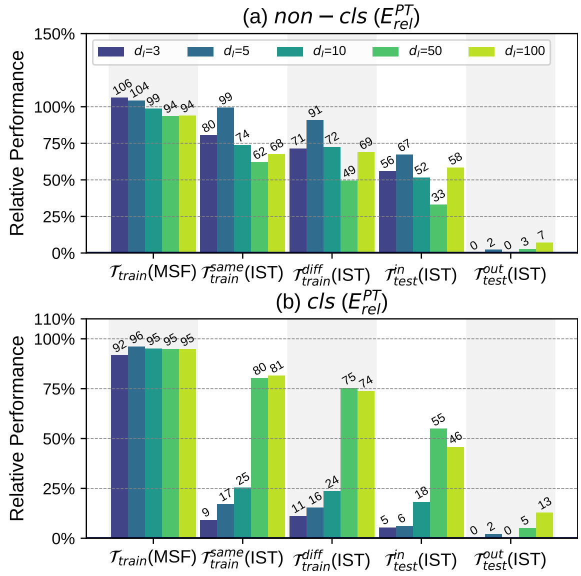


Figure 9: Relative performance of IPT with $\text{BART}_{\text{LARGE}}$ at different dimension d_I on *non-cls* and *cls* splits, comparing with prompt tuning.

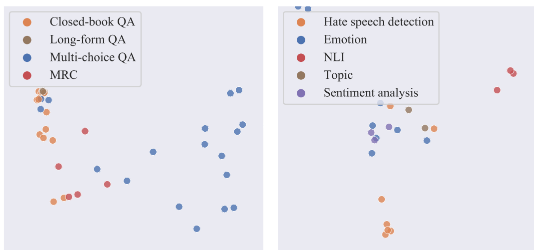
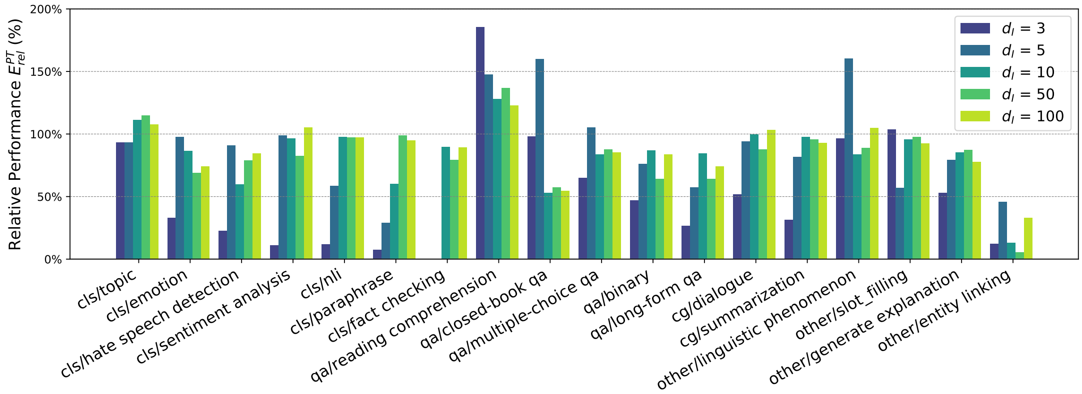


Figure 10: PCA plots of the intrinsic vectors learned during IST. We label points with different colors to represent its corresponding categories. Specifically, we show the clusters of fine-grained categories of QA (left) and text classification tasks (right). Without loss of generality, we choose the task split of *random* and $d_I = 100$.

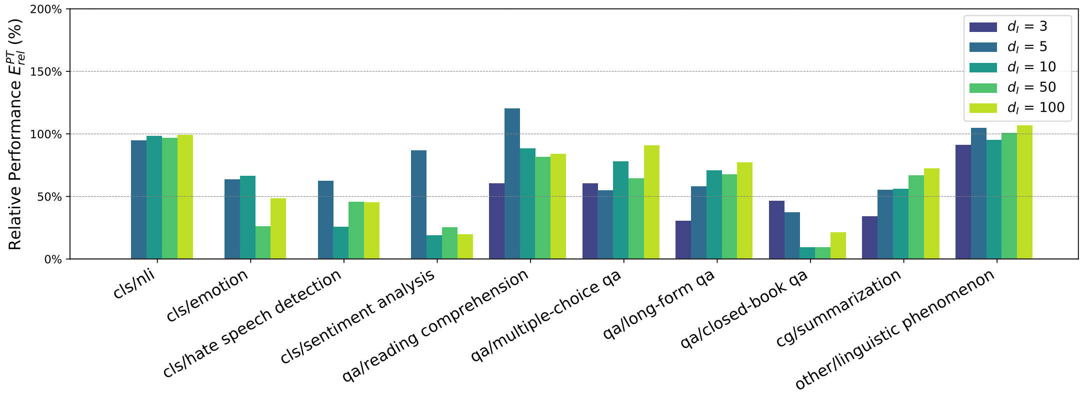
Tuning PLMs within intrinsic task subspaces. We have shown in Table 2 and appendix A.4 that IPT can improve tuning stability. We encourage future works to explore more methods to tune PLMs within low-dimensional intrinsic task subspaces, which may have more practical benefits such as avoiding over-parameterization and being greener to environments with fewer tunable parameters.

D Implementation Details

For all experiments, we adopt AdamW (Loshchilov and Hutter, 2019) as the optimizer. We train all



(a) Unseen-data Challenge ($\mathcal{T}_{\text{train}}^{\text{diff}}$(IST))



(b) Unseen-task Challenge ($\mathcal{T}_{\text{test}}$(IST))

Figure 11: We report $E_{\text{rel}}^{\text{PT}}$ of IPT at different d_I on tasks grouped by fine-grained task types under the (a) *unseen-data challenge* ($\mathcal{T}_{\text{train}}^{\text{diff}}$(IST)) and (b) *unseen-task challenge* ($\mathcal{T}_{\text{test}}$(IST)), respectively. The results come from the *random* task split.

models under the same environment of NVIDIA 32GB V100 GPU. We perform grid search on the combination of a series of learning rates ($\{1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}\}$) and batch sizes ($\{2, 4, 8\}$)[2], choose the best checkpoint using $\mathcal{D}_{\text{dev}}$, and evaluate it on $\mathcal{D}_{\text{test}}$. We set the max step to $10,000$ / $100,000$ and validate on $\mathcal{D}_{\text{dev}}$ every 100 / 1000 steps[3]. The ratio α is set to 200. During MSF, we only select the prompts that perform best on $\mathcal{D}_{\text{dev}}$ for each task to train the auto-encoder since we empirically found that involving other prompts leads to worse performance. The hyper-parameters of IST are chosen as the same as prompt tuning for fair comparisons.

For detailed model implementation, as mentioned in § 3.2, the projection function $\mathbf{Proj}(\cdot)$ is implemented with a one-layer feed-forward network and $\mathbf{Proj}_b(\cdot)$ is parameterized by a two-layer perceptron as follows:

$$\mathbf{Proj}_b(\mathbf{d}_i) = \mathbf{W}_2(\tanh(\mathbf{W}_1\mathbf{d}_i + \mathbf{b}_1)) + \mathbf{b}_2,$$

where $\mathbf{W}_1 \in \mathbb{R}^{d'_I \times d_I}$, $\mathbf{b}_1 \in \mathbb{R}^{d'_I}$, $\mathbf{W}_2 \in \mathbb{R}^{n \times d \times d'_I}$ and $\mathbf{b}_2 \in \mathbb{R}^{n \times d}$ are trainable parameters. d_I denotes the intrinsic dimension investigated in this paper. We set the inner hidden size d'_I of $\mathbf{Proj}_b$ to 768 for both BART$_{\text{BASE}}$ and BART$_{\text{LARGE}}$.

E Fine-grained Performance of IPT

In § 4, we evaluate the performance of IPT on 120 tasks and also divide them into cls. (classification) and non-cls. (non-classification) tasks to see the difference between these two types. Here we take a step further to investigate IPT at a more fine-grained level based on the task ontology of Ye et al. (2021). Specifically, we divide cls. tasks into 7 types (cls/topic, cls/emotion, cls/nli, cls/fact checking, cls/hate speech detection, cls/sentiment

[2] The numbers are chosen by pilot experiments on a random subset of tasks

[3] We found that prompt tuning empirically requires around $10\times$ more steps than fine-tuning to converge.

analysis, cls/paraphrase) and non-cls. tasks into 11 types (qa/reading comprehension, qa/closed-book qa, qa/multiple-choice qa, qa/binary, qa/long-form qa, cg/dialogue, cg/summarization, other/linguistic phenomenon, other/slot filling, other/generate explanation, other/entity linking). We show the relative performance compared with prompt tuning on *unseen-data challenge* ($\mathcal{T}_{\text{train}}^{\text{diff}}$(IST)) and *unseen-task challenge* ($\mathcal{T}_{\text{test}}$(IST)) in Figure 11, from which we can observe that IPT achieve obvious improvements compared to vanilla prompt tuning on some fine-grained types such as the qa/reading comprehension, which indicates that tuning PLMs within the intrinsic task subspace is promising to obtain certain benefits.

F Task Details

We list details for all the evaluated tasks in this paper in Table 7.

Table 7: The tasks evaluated in our experiments. We refer to Ye et al. (2021) for task ontology.

Ontology	Task Name	Reference
cls/sentiment analysis	glue-sst2	Socher et al. 2013
	imdb	Maas et al. 2011
	rotten_tomatoes	Pang and Lee 2005
cls/emotion	emo	Chatterjee et al. 2019
	tweet_eval-emoji	Barbieri et al. 2020
	tweet_eval-hate	Barbieri et al. 2020
	tweet_eval-irony	Barbieri et al. 2020
	tweet_eval-offensive	Barbieri et al. 2020
	tweet_eval-sentiment	Barbieri et al. 2020
	tweet_eval-stance_abortion	Barbieri et al. 2020
	tweet_eval-stance_atheism	Barbieri et al. 2020
	tweet_eval-stance_climate	Barbieri et al. 2020
	tweet_eval-stance_feminist	Barbieri et al. 2020
	tweet_eval-stance_hillary	Barbieri et al. 2020
cls/hate speech detection	ethos-disability	Mollas et al. 2020
	ethos-gender	Mollas et al. 2020
	ethos-national_origin	Mollas et al. 2020
	ethos-religion	Mollas et al. 2020
	ethos-sexual_orientation	Mollas et al. 2020
	hate_speech18	Davidson et al. 2017
	hatexplain	Mathew et al. 2020
cls/NLI	anli	Nie et al. 2020
	glue-mnli	Williams et al. 2018
	glue-qnli	Rajpurkar et al. 2016
	glue-rte	Dagan et al. 2005; Bar-Haim et al. 2006 Giampiccolo et al. 2007; Bentivogli et al. 2009
	glue-wnli	Faruqui and Das 2018
	scitail	Khot et al. 2018
	superglue-rte	Dagan et al. 2005; Bar-Haim et al. 2006 Giampiccolo et al. 2007; Bentivogli et al. 2009
cls/fact checking	climate_fever	Diggelmann et al. 2020
	kilt_fever	Thorne et al. 2018
	liar	Wang 2017
cls/paraphrase	glue-qqp	(link)
	medical_questions_pairs	McCreery et al. 2020
	paws	Zhang et al. 2019
cls/topic	ag_news	Gulli (link)
	dbpedia_14	Lehmann et al. 2015
cls/other	ade_corpus_v2-classification	Gurulingappa et al. 2012
	discovery	Sileo et al. 2019
	glue-cola	Warstadt et al. 2019
	google_wellformed_query	Faruqui and Das 2018
	sms_spam	Almeida et al. 2011
	superglue-wic	Pilehvar and Camacho-Collados 2019
	superglue-wsc	Levesque et al. 2012
	wiki_qa	Yang et al. 2015
qa/closed-book qa	freebase_qa	Jiang et al. 2019
	jeopardy	(link)
	kilt_hotpotqa	Yang et al. 2018
	kilt_nq	Kwiatkowski et al. 2019
	kilt_trex	Elsahar et al. 2018
	kilt_zsre	Levy et al. 2017
	lama-conceptnet	Petroni et al. 2019, 2020
	lama-google_re	Petroni et al. 2019, 2020
	lama-squad	Petroni et al. 2019, 2020
	lama-trex	Petroni et al. 2019, 2020
	numer_sense	Lin et al. 2020a
	search_qa	Dunn et al. 2017
	squad-no_context	Rajpurkar et al. 2016
	web_questions	Berant et al. 2013
qa/binary	boolq	Clark et al. 2019
	mc_taco	Zhou et al. 2019

Ontology	Task Name	Reference
qa/multiple-choice qa	ai2_arc	Clark et al. 2018
	aqua_rat	Ling et al. 2017
	codah	Chen et al. 2019
	commonsense_qa	Talmor et al. 2019
	cosmos_qa	Huang et al. 2019
	dream	Saha et al. 2018
	hellaswag	Zellers et al. 2019
	math_qa	Amini et al. 2019
	openbookqa	Mihaylov et al. 2018
	qasc	Khot et al. 2020
	quail	Rogers et al. 2020
	quarel	Tafjord et al. 2019a
	quartz-no_knowledge	Tafjord et al. 2019b
	quartz-with_knowledge	Tafjord et al. 2019b
	race-high	Lai et al. 2017
	race-middle	Lai et al. 2017
	social_i_qa	Sap et al. 2019
	superglue-copa	Gordon et al. 2012
	superglue-multirc	Khashabi et al. 2018
	swag	Zellers et al. 2018
	wino_grande	Sakaguchi et al. 2020
qa/long-form qa	eli5-askh	Fan et al. 2019
	eli5-asks	Fan et al. 2019
	eli5-eli5	Fan et al. 2019
qa/MRC	adversarialqa	Bartolo et al. 2020
	biomrc	Pappas et al. 2020
	quoref	Dasigi et al. 2019
	ropes	Lin et al. 2019
	superglue-record	Zhang et al. 2018
cg/summarization	gigaword	Napoles et al. 2012
	multi_news	Fabbri et al. 2019
	samsum	Gliwa et al. 2019
	xsum	Narayan et al. 2018
cg/dialogue	empathetic_dialogues	Rashkin et al. 2019
	kilt_wow	Dinan et al. 2019
cg/other	spider	Yu et al. 2018
	wiki_bio	Lebret et al. 2016
	wiki_split	Botha et al. 2018
	wikisql	an 2017
other/linguistic phenomenon	blimp-anaphor_gender_agreement	Warstadt et al. 2020
	blimp-ellipsis_n_bar_1	Warstadt et al. 2020
	blimp-sentential_negation_npi_scope	Warstadt et al. 2020
other/generate explanation	cos_e	Rajani et al. 2019
other/slot_filling	ade_corpus_v2-dosage	Gurulingappa et al. 2012
	ade_corpus_v2-effect	Gurulingappa et al. 2012
other/entity linking	kilt_ay2	Hoffart et al. 2011
other/other	acronym_identification	Pouran Ben Veyseh et al. 2020
	art	Bhagavatula et al. 2020
	aslg_pc12	Othman and Jemni 2012
	break-QDMR	Wolfson et al. 2020
	break-QDMR-high-level	Wolfson et al. 2020
	common_gen	Lin et al. 2020b
	crawl_domain	Zhang et al. 2020
	crows_pairs	Nangia et al. 2020
	definite_pronoun_resolution	Rahman and Ng 2012
	e2e_nlg_cleaned	Dušek et al. 2020, 2019
	limit	Manotas et al. 2020
	piqa	Bisk et al. 2020
	proto_qa	Boratko et al. 2020
	qa_srl	He et al. 2015