# A Few More Examples May Be Worth Billions of Parameters

**Yuval Kirstain**$^{\tau}$ **Patrick Lewis**$^{\lambda\mu}$ **Sebastian Riedel**$^{\lambda\mu}$ **Omer Levy**$^{\tau\mu}$

$^{\tau}$ Tel-Aviv University
$^{\lambda}$ University College London
$^{\mu}$ Meta AI

## Abstract

We investigate the dynamics of increasing the number of model parameters versus the number of labeled examples across a wide variety of tasks. Our exploration reveals that while scaling parameters consistently yields performance improvements, the contribution of additional examples highly depends on the task's format. Specifically, in open question answering tasks, enlarging the training set does not improve performance. In contrast, classification, extractive question answering, and multiple choice tasks benefit so much from additional examples that collecting a few hundred examples is often "worth" billions of parameters. We hypothesize that unlike open question answering, which involves recalling specific information, solving strategies for tasks with a more restricted output space transfer across examples, and can therefore be learned with small amounts of labeled data.[1]

## 1 Introduction

Recent work on few-shot learning for natural language tasks explores the dynamics of scaling up either the number of model parameters (Brown et al., 2020) or labeled examples (Le Scao and Rush, 2021), while controlling for the other variable by setting it to a constant. For example, Brown et al. (2020) focus on in-context learning from roughly 32 to 64 examples, a practice that was adopted by fine-tuning approaches as well (Schick and Schütze, 2021b; Gao et al., 2021b; Tam et al., 2021); however, there are many practical few-shot scenarios where *hundreds* of examples can be collected at a relatively low effort.[2] Other work experiments with single-size models (Schick and Schutze, 2020; Ram et al., 2021; Le Scao and Rush, 2021; Gao et al., 2021b), even though larger (or smaller) models may exhibit different behavior. Furthermore, much of the literature focuses on classification tasks (Schick and Schütze, 2021a; Gao et al., 2021b; Le Scao and Rush, 2021), leaving it unclear whether their conclusions generalize to tasks with less restricted output spaces.

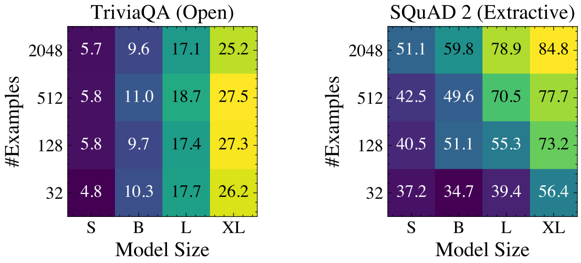


Figure 1: Open QA tasks (e.g. TriviaQA) benefit from additional parameters exclusively, while extractive QA tasks (e.g. SQuAD 2) benefit from both larger models and more labeled data.

In this paper, we conduct a systematic exploration of few-shot learning for language tasks, where we investigate the dynamics of increasing the number of model parameters (using different sizes of the self-supervised T5 (Raffel et al., 2020)) versus the number of target-task labeled examples (from 32 to 2048) across a variety of tasks, including not only classification, but also extractive, multiple-choice, and open question answering. Overall, we evaluate 192 scenarios by training 7,680 models to control for hyperparameters and random seeds.

Our experiments show that, surprisingly, the contribution of additional parameters versus additional labeled examples highly depends on the *format* of the task. For open QA tasks, such as the open-domain version of Natural Questions (Kwiatkowski et al., 2019; Lee et al., 2019), which require the model to recall specific information seen during pretraining, *enlarging the training set does not improve performance*. By contrast, increasing the

[1] Our code is publicly available: https://github.com/yuvalkirstain/lm-evaluation-harness.

[2] In SQuAD (Rajpurkar et al., 2016), for example, the average annotation pace is around one minute per question, producing 480 examples in a single 8-hour workday.

*Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1017–1029
December 7-11, 2022 ©2022 Association for Computational Linguistics

number of model parameters results in substantial gains (see TriviaQA (Joshi et al., 2017) in Figure 1). Hence, when dealing with open QA, model parameters are of immense value, and cannot be replaced by increasing the number of labeled examples.

On the other hand, we observe a completely different trend for classification, extractive QA, and multiple-choice tasks. These tasks benefit from enlarging *both* the training set and the model (see SQuAD 2 (Rajpurkar et al., 2018) in Figure 1). We observe that hundreds of examples are often "worth" *billions* of parameters; T5-L fine-tuned on 4 times more data is roughly competitive with T5-XL, which has 4 times the number of parameters. Moreover, some tasks benefit so much from labeled examples, that collecting even 512 data points can make a fine-tuned T5-L (800M parameters) outperform GPT-3 (175B parameters).

Finally, we hypothesize that unlike open QA, formats with restricted output spaces have solving strategies (such as elimination) that can be learned from small amounts of labeled data. This hypothesis also provides a possible explanation as to why lean retrieve-and-read approaches (such as DrQA (Chen et al., 2017), ORQA (Lee et al., 2019), and DPR (Karpukhin et al., 2020))) appear to be more robust than multi-billion-parameter closed-book models (Roberts et al., 2020) when tested on non-overlapping data (Lewis et al., 2021).

## 2 Experiments

We describe the tasks (Section 2.1), models (Section 2.2), data regimes (Section 2.3), and implementation details (Section 2.4) of our systematic experiment suite. In total, we experiment with 12 tasks, 4 models, 4 data regimes (with 5 samples each), and 8 hyperparameter configurations; these amount to 7,680 trained models, evaluated across 192 task-model-data scenarios.

### 2.1 Datasets

We experiment with 12 datasets, divided into 4 broad types of *task formats*. The formats and their constituent tasks are described below.

**Classification** In classification tasks, the model is expected to read a given text and predict a single label from a small closed set, e.g. *yes* or *no*. We adopt classification tasks from the GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) benchmarks, namely: Recognizing Textual Entailment (RTE, Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), the Stanford Sentiment Treebank (SST-2, Socher et al., 2013), and BoolQ (Clark et al., 2019). We report accuracy for all classification datasets.

**Extractive Question Answering** In extractive QA, the model is given a passage and a question, and is then expected to produce an answer in the form of a span from the passage. We experiment with SQuAD 2 (Rajpurkar et al., 2018), HotpotQA (Yang et al., 2018), and DROP (Dua et al., 2019). Each of these datasets contains an additional "quirk" that makes it more challenging than the original SQuAD dataset (Rajpurkar et al., 2016), which popularized the extractive QA format: SQuAD 2 has unanswerable questions, HotpotQA provides multiple passages per question, and DROP contains many arithmetic questions whose answer is not strictly extractive, but can be derived from a set of spans in the given passage. For all extractive QA datasets we report token-wise F1.

**Multiple Choice** Multiple choice tasks provide the model with a question and several candidate answers, with the goal of selecting the correct one. We focus on three datasets in this format: the easy question set from the AI2 Reasoning Challenge (ARC-E, Clark et al., 2018), the Physical Interaction Question Answering dataset (PIQA, Bisk et al., 2020), and CommonsenseQA (Talmor et al., 2019). Unlike extractive QA, multiple choice tasks do *not* contain supporting evidence (a passage) for answering the question, and in contrast to classification, they have a different output space (candidate answers) for each example. We report accuracy for all multiple choice datasets.

**Open Question Answering** Open QA[3] datasets provide the model with just a question; no supporting evidence or closed candidate set is available. We experiment with open-domain versions of Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and SQuAD 1 (Rajpurkar et al., 2016). Our experiments here focus purely on the closed-book setting (Roberts et al., 2020), which does not allow models to retrieve text from an external corpus, restricting them to information stored in their parameters. For all open QA tasks we report F1 as the main metric.

[3] We deviate from the widely-used term *open-domain QA*, which describes the task, and use *open QA* instead to refer to the format, much like we use *extractive QA* rather than *reading comprehension*.

### 2.2 Models

The Text-to-Text Transfer Transformer (T5, Raffel et al., 2020) uses an encoder-decoder transformer architecture. It is pretrained on the task of generating masked-out spans over the Colossal Clean Crawled Corpus (C4), which contains 800GB of English-language text. We use version 1.1 of T5, which is not trained on any labeled data. Our experiments include the 77M (S), 250M (B), 800M (L), and 3B (XL) parameter variants of this model.

### 2.3 Training Data

While many publicly released datasets include an enormous number of labeled examples (Rajpurkar et al., 2018; Yang et al., 2018; Kwiatkowski et al., 2019), and recent work on few-shot learning focus on an extreme scenario in which less than one hundred examples are at hand (Brown et al., 2020; Schick and Schütze, 2021a; Gao et al., 2021b), we choose to simulate a broader set of practical scenarios where a small-to-medium dataset is available; in SQuAD (Rajpurkar et al., 2016), for example, the average annotation pace is around one minute per question, producing 480 examples in a single 8-hour workday. Therefore, we consider 4 dataset sizes for each task: 32, 128, 512, and 2048 examples. For each dataset size, we sample the relevant amount of examples five times using different seeds, thus creating 20 datasets for each task overall. We report the average score for each dataset size, thereby reducing the high variance associated with training on small datasets.

### 2.4 Implementation

**Code** For our implementation we extend EleutherAI's language model evaluation harness (Gao et al., 2021a) to allow fine-tuning and evaluating additional datasets and models.

**Prompts** We adopt the prompts used by Brown et al. (2020) and Khashabi et al. (2020), with minimal adaptations to T5 by adding a mask token followed by a period. Following Le Scao and Rush (2021), we use prompts in conjunction with fine-tuning.

**Decoding** We use greedy decoding for extractive and open QA tasks. For classification and multiple choice tasks, we compare the model's probability for each possible outcome, and predict the option with the highest probability. In BoolQ, for example, we compare $P(\text{“yes”}|x)$ (the probability of the positive class) with $P(\text{“no”}|x)$ (the probability of the negative class), where $x$ is the prompt containing the context and the question.

**Reproducibility and Hardware** While our evaluation suite is extensive, and therefore compute-heavy, the average runtime on the experiments is less than one hour, and can be executed with merely four 32GB V100 GPUs. [4] This allows one to *verify* the results by sampling a small subset of scenarios, and testing them in low-resource setup.

**Hyperparameters** To tune hyperparameters for fine-tuning, we split the available data into 75% training and 25% validation (e.g. 24 training examples and 8 validation when the dataset size is 32). For each case, we experiment with two learning rates ($5\text{e}{-5}$, $5\text{e}{-4}$) combined with linear decay, two weight decays ($0.001$, $0.1$), and two values for amount of steps ($512$, $2048$). The effective batch size is always 32 examples. Additionally, we use a dropout ratio of 0.1, gradient clipping is set to 1, and the amount of warmup steps is determined by the maximum between 10% of the training steps and 100. We evaluate each run after every epoch and choose the model with the lowest validation loss for extractive and open QA tasks, and highest validation accuracy for classification and multiple choice tasks.

## 3 Results

For each task in our experiment suite, we present a heatmap of the model's performance as a function of model size and the number of labeled examples. These heatmaps expose that most tasks benefit from both larger models *and* more training data, to a point where enlarging the dataset will result in similar gains as increasing the number of parameters. However, this trend does not apply to open QA tasks, whose performance *only* improves with additional model parameters (Section 3.1). Furthermore, we show that converting multiple choice datasets to the open QA format *disables* the benefits of additional training data, whereas converting in the opposite direction – from open QA to extractive or multiple-choice QA – *enables* models to improve with more examples (Section 3.2). Next, we describe a method for quantifying the relative benefit from parameters versus training data, which confirms the observed trends (Section 3.3).

[4] Most experiments require a single GPU, and only those that train billion parameter models require four.

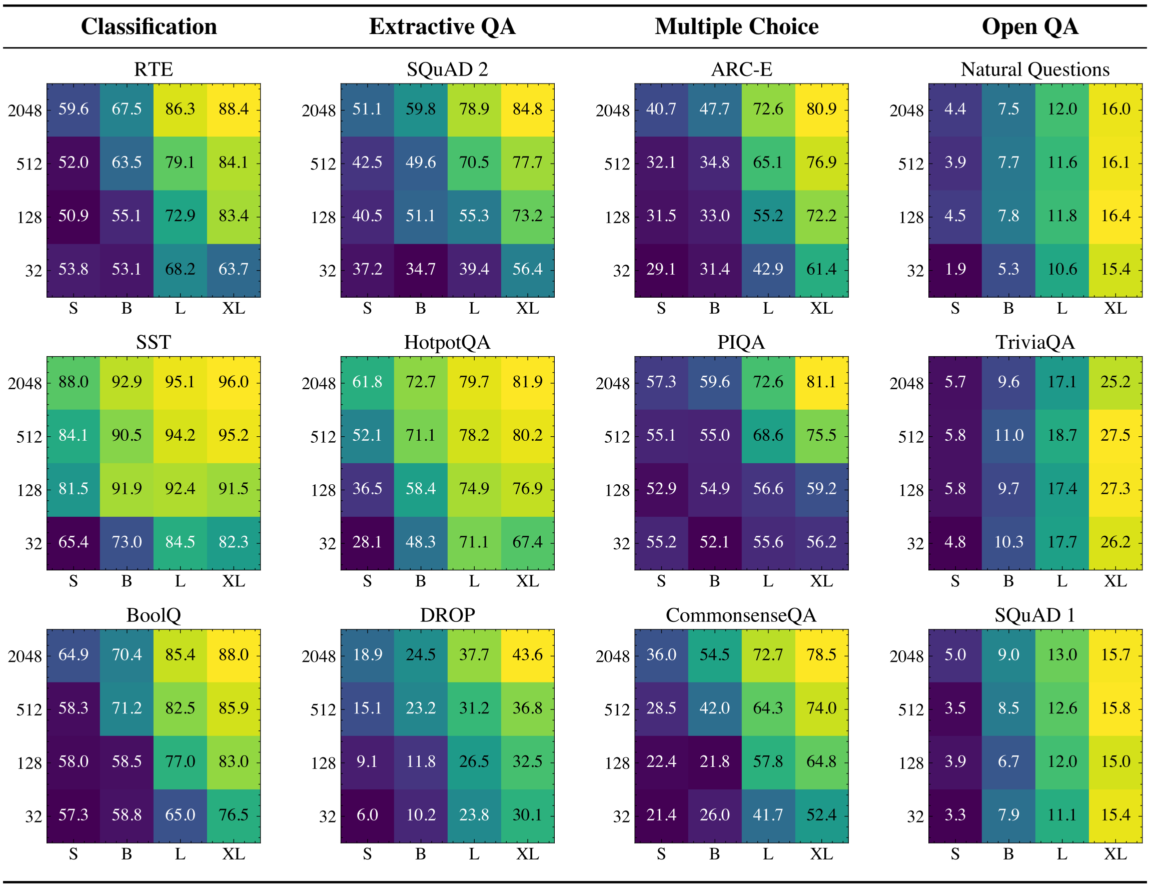


Figure 2: Each heatmap displays the model's performance (F1/accuracy) given its size in parameters (horizontal axis) and the number of labeled examples available during fine-tuning (vertical axis).

We then show that collecting a few hundred examples allows the much smaller T5-L to outperform GPT3, but not in open QA tasks, where the massive amount of parameters is the prime contributor (Section 3.4). Finally, we suggest a hypothesis to explain the observed trends (Section 3.5).

### 3.1 Main Trends

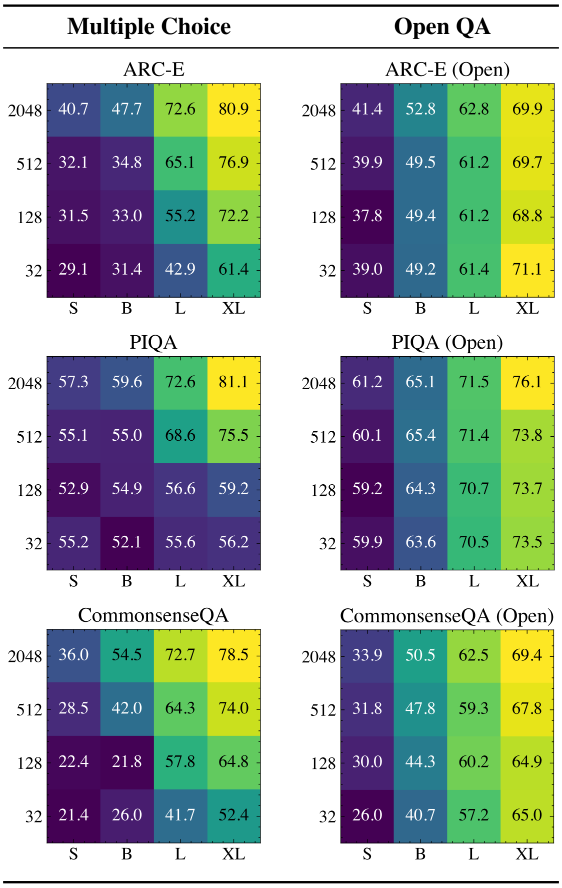
Figure 2 shows performance as a function of model size and dataset size per task.[5] Visualizing the results via heatmaps highlights two patterns: (1) *diagonal gradients*, where performance significantly improves along both axes (though not necessarily equally), and (2) *horizontal gradients*, where performance improves almost exclusively along the horizontal (model size) axis. We observe that all three open QA datasets exhibit horizontal gradients, while the remaining datasets follow the diagonal patterns. We do not observe *vertical gradients* at all, indicating that enlarging the model's size is consistently beneficial.

Consider TriviaQA, for example (Figure 2, right column, second row); performance approximately doubles when switching models from T5-B to T5-L (and from T5-L to T5-XL), but changes by less than 2 points when increasing the dataset from 32 examples to 2048. On the other hand, in the classification task SST-2 (Figure 2, left column, second row), annotating 128 examples rather than 32 examples results in double-digit improvements for T5-S and T5-B, and in significant gains for larger models as well. Here, data-driven improvements coincide with parameter-driven improvements, and increasing either factor typically boosts performance. Moreover, the diagonal gradients show that in many cases a model trained on more data can "catch up" with a larger model. This trend is particularly striking when comparing T5-L with T5-XL, where training the smaller model (T5-L) on 4 times more data is almost always competitive with the larger model (T5-XL).

### 3.2 Same Dataset, Different Format

While a clear dichotomy arises from Section 3.1 with respect to format, it might also result from the

[5] The results are available in tabular form in Appendix A.



ARC-E

| | S | B | L | XL |
|---|---|---|---|---|
| 2048 | 40.7 | 47.7 | 72.6 | 80.9 |
| 512 | 32.1 | 34.8 | 65.1 | 76.9 |
| 128 | 31.5 | 33.0 | 55.2 | 72.2 |
| 32 | 29.1 | 31.4 | 42.9 | 61.4 |

ARC-E (Open)

| | S | B | L | XL |
|---|---|---|---|---|
| 2048 | 41.4 | 52.8 | 62.8 | 69.9 |
| 512 | 39.9 | 49.5 | 61.2 | 69.7 |
| 128 | 37.8 | 49.4 | 61.2 | 68.8 |
| 32 | 39.0 | 49.2 | 61.4 | 71.1 |

PIQA

| | S | B | L | XL |
|---|---|---|---|---|
| 2048 | 57.3 | 59.6 | 72.6 | 81.1 |
| 512 | 55.1 | 55.0 | 68.6 | 75.5 |
| 128 | 52.9 | 54.9 | 56.6 | 59.2 |
| 32 | 55.2 | 52.1 | 55.6 | 56.2 |

PIQA (Open)

| | S | B | L | XL |
|---|---|---|---|---|
| 2048 | 61.2 | 65.1 | 71.5 | 76.1 |
| 512 | 60.1 | 65.4 | 71.4 | 73.8 |
| 128 | 59.2 | 64.3 | 70.7 | 73.7 |
| 32 | 59.9 | 63.6 | 70.5 | 73.5 |

CommonsenseQA

| | S | B | L | XL |
|---|---|---|---|---|
| 2048 | 36.0 | 54.5 | 72.7 | 78.5 |
| 512 | 28.5 | 42.0 | 64.3 | 74.0 |
| 128 | 22.4 | 21.8 | 57.8 | 64.8 |
| 32 | 21.4 | 26.0 | 41.7 | 52.4 |

CommonsenseQA (Open)

| | S | B | L | XL |
|---|---|---|---|---|
| 2048 | 33.9 | 50.5 | 62.5 | 69.4 |
| 512 | 31.8 | 47.8 | 59.3 | 67.8 |
| 128 | 30.0 | 44.3 | 60.2 | 64.9 |
| 32 | 26.0 | 40.7 | 57.2 | 65.0 |

Figure 3: Converting multiple choice tasks (left column) to open QA (right column) changes the scaling dynamics, replacing diagonal gradients (performance improves with more parameters and more data) to horizontal gradients (performance improves almost exclusively with more parameters).

fact that the different datasets were collected and annotated using different methodologies. Can we conduct a more controlled experiment, which uses the same dataset but in different formats?

We first take the three multiple choice datasets (ARC-E, PIQA, and CommonsenseQA) and convert them into the open QA format by excluding the candidate answers from the input.[6] Figure 3 shows that the diagonal gradients clearly seen in the multiple choice format are replaced with horizontal gradients similar to those of other open QA datasets.

We also examine data conversion in the opposite direction, by using multiple choice and extractive QA versions of Natural Questions.[7] Here we control for the change in format by decoding the multiple choice models as we do for extractive and open QA tasks and report F1. Figure 4 shows that while the open QA heatmap displays largely horizontal gradients, both extractive QA and multiple choice heatmaps follow the diagonal patterns. Unlike the original open-domain Natural Questions dataset, we do observe some minor improvement along the data axis in this entity-focused version, but analyzing the data reveals that this stems from an increase in example overlap (Lewis et al., 2021), with 11.7% of test-set answers appearing in the 2048-example training sets, compared to 8.5% in the original. Overall, both experiments' results indicate that the task's format directly impacts whether more labeled data will improve performance or not.

## 3.3 Quantifying the Relative Impact of Parameters versus Examples

For many tasks, both additional model parameters and labeled examples can improve performance. However, it is not always clear *how much* each factor contributes to greater performance gains with respect to the other. To quantify the importance of increasing parameters versus examples, we compute a regression-based metric using the numerical results in a given heatmap. Specifically, we train the following linear regression model for each heatmap:

$$y = \alpha_m x_m + \alpha_d x_d + b$$

where $y$ is the model's performance on the task, $x_m$ is the normalized number of model parameters (S is 1, B is 2, L is 3, and XL is 4), and $x_d$ is the normalized number of dataset examples (32 is 1, 128 is 2, 512 is 3, and 2048 is 4). The regression coefficients $\alpha_m, \alpha_d$ are scalars, learnt for each task, which are then normalized to measure the relative impact of each axis (parameters versus examples):

$$I_m = \frac{|\alpha_m|}{|\alpha_m| + |\alpha_d|}$$

When $0 < I_m < 0.5$, additional examples are greater contributors to performance gains, while

[6] We control for the inference method by selecting the most probable answer candidate, rather than applying greedy decoding. Thus, the only difference between each pair of datasets is whether or not the candidates appear in the input.

[7] The original Natural Question dataset (Kwiatkowski et al., 2019) is in the extractive QA format; specifically, we use the version in the 2019 MRQA Shared Task (Fisch et al., 2019). We filter the dataset to include only named entity answers that were recognized using an off-the-shelf OntoNotes Named Entity Recognition model from spaCy (Hovy et al., 2006; Honnibal et al., 2020), and suggest them as candidate answers alongside entities of the same type that appear in the background passage.

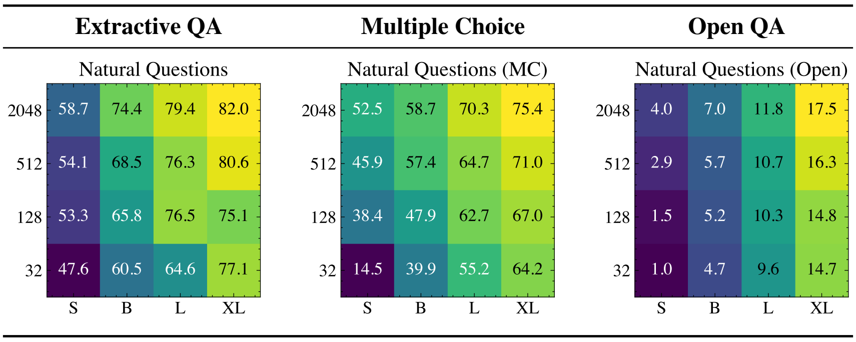


Figure 4: Converting Natural Questions (Kwiatkowski et al., 2019) from its open QA format (right) to multiple choice (middle) and extractive QA (left) changes the scaling dynamics, replacing horizontal gradients (performance improves almost exclusively with more parameters) with diagonal gradients (performance improves with more parameters and more data).

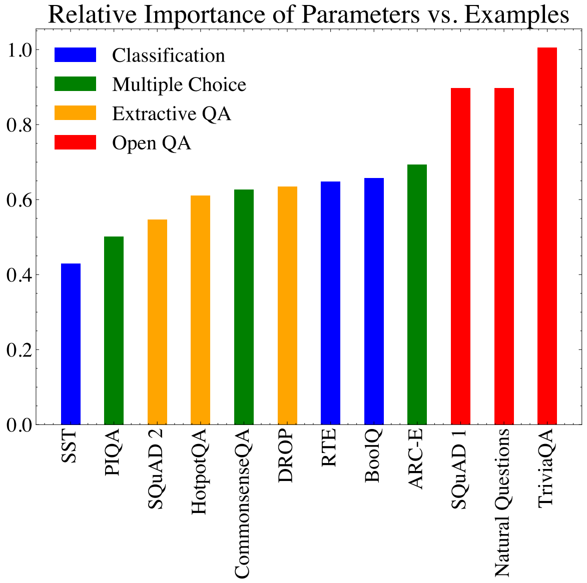


Figure 5: The relative importance of parameters versus examples ($I_m$), as computed via regression over each tasks's heatmap. Higher values indicate more dependence on parameters and less on labeled data.

$0.5 < I_m < 1.0$ indicates that model parameters have higher relative importance.

Figure 5 shows that most tasks lie between $0.4 < I_m < 0.7$, with model parameters responsible for most performance improvements, but with significant improvements arising from labeled data as well. However, all open QA tasks deviate from this interval, and exhibit $I_m$ values of 0.9 and above, indicating that increased model parameters is almost exclusively responsible for better performance.

### 3.4 Comparison with Massive Models

While models can benefit from both parameters and labeled data in many tasks, scaling up language models to hundreds of billions of parameters may restrict the ability to fine-tune, as GPT3-scale models are typically available only as a service to most practitioners and researchers (Brown et al., 2020). Given this data-parameter trade-off, how many labeled examples are 175B parameters worth?

We compare our results of T5-L (800M parameters) fine-tuned on various dataset sizes to those of GPT3 (175B, over 200 times larger than T5-L) using in-context learning, as reported by Brown et al. (2020).[8] Table 1 shows a wide performance gap between GPT3 and T5-L on open QA datasets, which cannot be bridged by additional labeled examples, as observed in our main experiments. However, for classification and extractive QA tasks, even a few hundred labeled examples are often enough for T5-L to catch up with GPT3's performance and even exceed it. In BoolQ, for example, just collecting 96 additional examples is tantamount to adding 200 times more parameters to the model. This result demonstrates that while performance may improve along the parameter axis in classification and extractive QA tasks, a small amount of labeled training data can also go a long way.

### 3.5 Discussion

Why does changing the task's input format have such a dramatic effect on the training dynamics? We conjecture that the format changes described in our experiments, which effectively remove information from the input, force the models to supplement that information with knowledge stored in its parameters. For example, when asking *What is the capital of Micronesia?* in the open QA format, the

[8] Our work has 6 datasets that properly overlap with the original GPT3 paper. ARC-E and PIQA are also used to evaluate GPT3, but in the open QA format.

| Model | #Examples | Classification | | Extractive QA | | Open QA | |
|---|---|---|---|---|---|---|---|
| | | RTE | BoolQ | SQuAD 2 | DROP | NQ | TriviaQA |
| T5-L | 32 | 68.2 | 65.0 | 39.4 | 23.8 | 5.1 | 12.0 |
| | 128 | **72.9** | 77.0 | 55.3 | 26.5 | 5.8 | 11.5 |
| | 512 | **79.1** | **82.5** | **70.5** | 31.2 | 5.6 | 12.1 |
| | 2048 | **86.3** | **85.4** | **78.9** | **37.7** | 6.3 | 10.7 |
| **GPT3** | ≤64 | 72.9 | 77.5 | 69.8 | 36.5 | 29.9 | 71.2 |

Table 1: A comparison between GPT3 (with in-context learning, as reported by Brown et al. (2020)) and T5-L. Figures in bold represent T5-L configurations that outperform GPT3. For a fair comparison with Brown et al. (2020), we report accuracy (exact match) for open QA tasks in this table.

model is required to know that the answer is *Palikir* by encountering the fact during pretraining or fine-tuning on a paraphrase of the same question. In contrast, if the same question is asked in the multiple choice format, and the options are *(1) Rome, (2) Tokyo, (3) Yaren, (4) Palikir*, the model can easily eliminate the more frequently-mentioned capitals of Rome and Tokyo, and then guess between the two remaining options, *Yaren* (the capital of neighboring Nauru) and *Palikir* (the correct answer). A similar example can be constructed for extractive QA, where the vast majority of passage spans can be pruned a priori, leaving only a handful of named entities as more likely candidates. We hypothesize that answering strategies, such as elimination, can indeed be learned from small-medium training sets, while actual new facts cannot, unless there is significant train-test overlap (Lewis et al., 2021).

A practical corollary of this hypothesis is that if one can modify a target task from the open QA format to one with a more limited output space, à la multiple choice or extractive QA, they would unlock the ability to trade data for parameters. Instead of relying on massive pretrained language models, which can only be used as a service, one could achieve competitive and even superior results with a much smaller model, given a relatively small dataset of several hundred labeled examples. Retrieve-and-read approaches successfully demonstrate this notion by decomposing open QA into two separate classification and extractive QA subproblems (Chen et al., 2017; Lee et al., 2019; Karpukhin et al., 2020), and may possibly be applied to few-shot scenarios in additional tasks via more general retrieve-and-generate models such as RAG (Lewis et al., 2020).

## 4 Related Work

Few-shot learning has been a subject of interest for several decades (Thrun and Pratt, 1998; Fink, 2005; Li et al., 2006; Vinyals et al., 2016; Jiang et al., 2018). Within NLP, “few-shot learning" has traditionally focused on quickly learning new classes and domains within the context of a single task (see Yin (2020) for a recent survey). Recently, there has been a surge of interest in few-shot learning, following the release of GPT3 (Brown et al., 2020). Here, the few-shot learning paradigm has shifted subtly, and refers to building models capable of tackling a range of standard NLP tasks, albeit using very restricted training sets, usually sub-sampled from the full training set. A great deal of work has recently been produced in this area, and we provide a necessarily incomplete summary below.

**In-Context Learning** In-context learning (Brown et al., 2020) generally refers to adapting to a task by providing training examples as additional textual input, without performing gradient-based updates. This technique imposes a limit on size of the training dataset due to context length limits. Recent work from Liu et al. (2021a) and Lu et al. (2021) demonstrate that the choice of in-context training examples, and the order in which they are presented have large effects on performance.

**Prompt-Based Learning** Prompting refers to providing additional input to a model designed to help it to produce correct outputs. Typically, these take the form of textual templates used to form cloze questions, and have been used in a variety of settings, such as probing (Petroni et al., 2019) and zero-shot learning (Radford et al., 2019). Prompts can be used in conjunction with fine-tuning, which has been shown to improve results in a number of works (Schick and Schutze, 2020, 2021; Schick and Schütze, 2021a; Gao et al., 2021b; Le Scao and Rush, 2021; Tam et al., 2021). We adopt this technique in our experiments and adapt the models using prompt-based fine-tuning.

**Prompt Engineering** Models may be sensitive to the choice of prompt (especially without finetuning), and a number of works attempt to optimize the prompt for the task at hand (Jiang et al., 2020; Shin et al., 2020). Recently, a number of works have also proposed generalizing prompts to include task-specific parameters and embeddings, typically learnt via gradient descent while keeping parts or all of the model's parameters frozen (Houlsby et al., 2019; Liu et al., 2021b; Zhong et al., 2021; Qin and Eisner, 2021; Li and Liang, 2021; Lester et al., 2021; Logan et al., 2021). While these techniques can improve results for frozen models, they generally do not outperform fine-tuning the whole model (Lester et al., 2021), hence we choose to focus on full-model finetuning with standard prompts in our experiments.

**Few-Shot Learning Analysis** Closest to our contribution are works placing an emphasis on the analysis of few-shot model behaviour, rather than focusing on schemes to improve performance. Le Scao and Rush (2021) quantify the benefit of prompting in few-shot learning, and Perez et al. (2021) critically discuss the difficulty of model selection and very low dataset sizes in few-shot learning. Our work is complementary, exploring the relationship between scale, dataset size, and task openendedness.

**Task Formats** Another important aspect of our work is the investigation of learning as a function of task format. Related work in this area includes research investigating reformulating a task into a different format, such as reducing tasks to NLI (White et al., 2017; Wang et al., 2018) or reading comprehension (Levy et al., 2017; Wu et al., 2020), or even reducing all tasks to a single format (Kumar et al., 2016; McCann et al., 2018). A related line of work seeks to understand tasks and datasets by changing or removing parts of the input, and, in-so-doing, changing the task format. Examples include hypothesis-only NLI baselines (Gururangan et al., 2018; Poliak et al., 2018), and document-only baselines in Reading Comprehension (Kaushik and Lipton, 2018; Sugawara et al., 2020). We also change the available input to a model for a given task, effectively changing the task format, while keeping the targets unchanged. We do this to measure the effect of the open-endedness of a task on sample complexity for differently sized models.

## 5 Conclusions

In this work, we present an empirical investigation on the relationships between (1) a task's format, (2) the number of labeled examples available for said task, and (3) the number of parameters the model tackling the task has. Through our extensive experiments, we determine that task format greatly affects the relative performance improvement that can be expected from increased training set size and parameter count. For tasks that do not require the recollection of specific external information – i.e. classification, multiple choice, and extractive QA – we find that more labeled data and larger models both reliably improve performance. In fact, for some of these tasks, adding a few hundred labeled examples is *more* beneficial than scaling up the model size by billions of parameters. It seems then, from a practitioner's perspective, that for many tasks where data is very sparse, the tried-and-true strategy of simply collecting more training data will often be a more effective strategy than attempting to scale to larger, more computationally-demanding models. However, the picture is very different for open QA tasks; for such tasks, we find that increasing the size of the training data barely improves performance, leaving parameter inflation as the only reliable approach to improve accuracy. Finally, we provide a hypothesis to explain these results and conclude with a practical corollary – when possible, changing the format from open QA into a more "self-contained" one will allow labeled data to bridge performance gaps between moderately-sized models and much larger ones.

## 6 Limitations

This work has two main limitations. First, we mostly experiment with different variants of T5, but do not repeat the full experiment suite on other model families. While we did conduct preliminary experiments on GPT-2 and GPT-J, we found that T5 models provide significantly stronger baselines when controlling for the number of parameters. Another limitation is that while the set of tasks is diverse, it is not exhaustive; in particular, we do not explore tasks that require generating longer sequences of text, such as summarization. Having said that, our experiment suite is extensive and costly as-is, and it is not clear whether the financial and environmental costs of expanding our experiments to encapsulate further model families and tasks can be justified.

## Acknowledgments

We thank Avia Efrat and Ori Ram for valuable feedback and discussions.

## References

Roy Bar-Haim, Ido Dagan, Bill Dolan, L. Ferro, Danilo Giampiccolo, and B. Magnini. 2006. The second pascal recognising textual entailment challenge.

L. Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The sixth pascal recognizing textual entailment challenge. In *TAC*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*.

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

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. 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*, abs/1803.05457.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Michael Fink. 2005. Object classification from a single example utilizing class relevance metrics. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

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

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021a. A framework for few-shot language model evaluation.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021b. 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, B. Magnini, Ido Dagan, and W. Dolan. 2007. The third pascal recognizing textual entailment challenge. In *ACL-PASCAL@ACL*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

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*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: The 90In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, page 57–60, USA. Association for Computational Linguistics.

Xiang Jiang, Mohammad Havaei, Gabriel Chartrand, Hassan Chouaib, Thomas Vincent, Andrew Jesson, Nicolas Chapados, and Stan Matwin. 2018. On the importance of attention in meta-learning for few-shot text classification. *CoRR*, abs/1806.00852.

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.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. 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.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1378–1387. JMLR.org.

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*, 7:452–466.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

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

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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Fei-Fei Li, R. Fergus, and P. Perona. 2006. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:594–611.

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.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, L. Carin, and Weizhu Chen. 2021a. What makes good in-context examples for gpt-3? *ArXiv*, abs/2101.06804.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/2103.10385.

IV Robert L. Logan, Ivana Balavzevi'c, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models. *ArXiv*, abs/2106.13353.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *CoRR*, abs/2104.08786.

Bryan McCann, N. Keskar, Caiming Xiong, and R. Socher. 2018. The natural language decathlon: Multitask learning as question answering. *ArXiv*, abs/1806.08730.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *CoRR*, abs/2105.11447.

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.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. *CoRR*, abs/2104.06599.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-shot question answering by pretraining span selection. 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 3066–3079, Online. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Timo Schick and H. Schutze. 2020. Few-shot text generation with pattern-exploiting training. *ArXiv*, abs/2012.11926.

Timo Schick and H. Schutze. 2021. It's not just size that matters: Small language models are also few-shot learners. *ArXiv*, abs/2009.07118.

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.

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.

Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. 2020. Assessing the benchmarking capacity of machine reading comprehension datasets. In *AAAI*, pages 8918–8927. AAAI Press.

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.

Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. *ArXiv*, abs/2103.11955.

Sebastian Thrun and Lorien Pratt. 1998. *Learning to Learn: Introduction and Overview*, page 3–17. Kluwer Academic Publishers, USA.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 996–1005, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online. 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.

Wenpeng Yin. 2020. Meta-learning for few-shot natural language processing: A survey. *CoRR*, abs/2007.09604.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: learning vs. learning to recall. *CoRR*, abs/2104.05240.

## A Tabular Results

Table 2 provides the results from our main experiment (Section 3, Figure 2) in tabular form.

| **Model** | **#Examples** | **Classification** | | | **Extractive QA** | | | **Multiple Choice** | | | **Open QA** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | RTE | SST | BoolQ | SQuAD2 | HPQA | DROP | ARC-E | PIQA | CSQA | NQs | TQA | SQuAD1 |
| **T5-S** | 32 | 53.8 | 65.4 | 57.3 | 37.2 | 28.1 | 6.0 | 29.1 | 55.2 | 21.4 | 1.9 | 4.8 | 3.3 |
| | 128 | 50.9 | 81.5 | 58.0 | 40.5 | 36.5 | 9.1 | 31.5 | 52.9 | 22.4 | 4.5 | 5.8 | 3.9 |
| | 512 | 52.0 | 84.1 | 58.3 | 42.5 | 52.1 | 15.1 | 32.1 | 55.1 | 28.5 | 3.9 | 5.8 | 3.5 |
| | 2048 | 59.6 | 88.0 | 64.9 | 51.1 | 61.8 | 18.9 | 40.7 | 57.3 | 36.0 | 4.4 | 5.7 | 5.0 |
| **T5-B** | 32 | 53.1 | 73.0 | 58.8 | 34.7 | 48.3 | 10.2 | 31.4 | 52.1 | 26.0 | 5.3 | 10.3 | 7.9 |
| | 128 | 55.1 | 91.9 | 58.5 | 51.1 | 58.4 | 11.8 | 33.0 | 54.9 | 21.8 | 7.8 | 9.7 | 6.7 |
| | 512 | 63.5 | 90.5 | 71.2 | 49.6 | 71.1 | 23.2 | 34.8 | 55.0 | 42.0 | 7.7 | 11.0 | 8.5 |
| | 2048 | 67.5 | 92.9 | 70.4 | 59.8 | 72.7 | 24.5 | 47.7 | 59.6 | 54.5 | 7.5 | 9.6 | 9.0 |
| **T5-L** | 32 | 68.2 | 84.5 | 65.0 | 39.4 | 71.1 | 23.8 | 42.9 | 55.6 | 41.7 | 10.6 | 17.7 | 11.1 |
| | 128 | 72.9 | 92.4 | 77.0 | 55.3 | 74.9 | 26.5 | 55.2 | 56.6 | 57.8 | 11.8 | 17.4 | 12.0 |
| | 512 | 79.1 | 94.2 | 82.5 | 70.5 | 78.2 | 31.2 | 65.1 | 68.6 | 64.3 | 11.6 | 18.7 | 12.6 |
| | 2048 | 86.3 | 95.1 | 85.4 | 78.9 | 79.7 | 37.7 | 72.6 | 72.6 | 72.7 | 12.0 | 17.1 | 13.0 |
| **T5-XL** | 32 | 63.7 | 82.3 | 76.5 | 56.4 | 67.4 | 30.1 | 61.4 | 56.2 | 52.4 | 15.4 | 26.2 | 15.4 |
| | 128 | 83.4 | 91.5 | 83.0 | 73.2 | 76.9 | 32.5 | 72.2 | 59.2 | 64.8 | 16.4 | 27.3 | 15.0 |
| | 512 | 84.1 | 95.2 | 85.9 | 77.7 | 80.2 | 36.8 | 76.9 | 75.5 | 74.0 | 16.1 | 27.5 | 15.8 |
| | 2048 | 88.4 | 96.0 | 88.0 | 84.8 | 81.9 | 43.6 | 80.9 | 81.1 | 78.5 | 16.0 | 25.2 | 15.7 |

Table 2: The performance (F1/accuracy) of different T5 models fine-tuned on different training set sizes, across 12 different datasets.