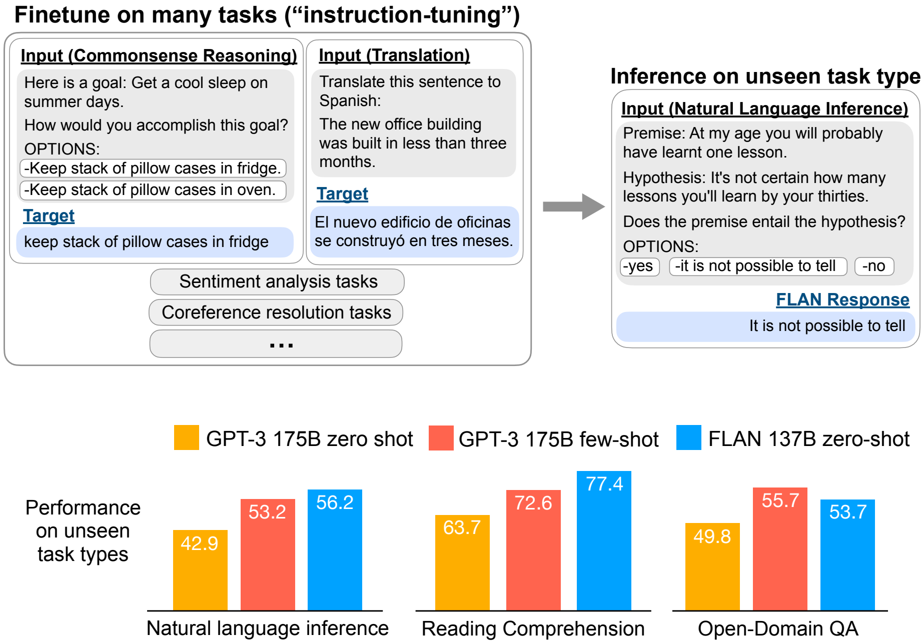
FINETUNED LANGUAGE MODELS ARE ZERO-SHOT LEARNERS

Jason Wei[*] **Maarten Bosma**[*] **Vincent Y. Zhao**[*] **Kelvin Guu**[*] **Adams Wei Yu**
Brian Lester **Nan Du** **Andrew M. Dai** **Quoc V. Le**
Google Research

ABSTRACT

This paper explores a simple method for improving the zero-shot learning abilities of language models. We show that *instruction tuning*—finetuning language models on a collection of tasks described via instructions—substantially boosts zero-shot performance on unseen tasks.

We take a 137B parameter pretrained language model and instruction-tune it on over 60 NLP tasks verbalized via natural language instruction templates. We evaluate this instruction-tuned model, which we call FLAN, on unseen task types. FLAN substantially improves the performance of its unmodified counterpart and surpasses zero-shot 175B GPT-3 on 19 of 25 tasks that we evaluate. FLAN even outperforms few-shot GPT-3 by a large margin on ANLI, RTE, BoolQ, AI2-ARC, OpenbookQA, and StoryCloze. Ablation studies reveal that number of tasks and model scale are key components to the success of instruction tuning.


Finetune on many tasks ("instruction-tuning")
Input (Commonsense Reasoning)
Target
Input (Translation)
Target
Sentiment analysis tasks
Coreference resolution tasks
...
Inference on unseen task type
Input (Natural Language Inference)
FLAN Response
GPT-3 175B zero shot
GPT-3 175B few-shot
FLAN 137B zero-shot
Performance on unseen task types
42.9
53.2
56.2
63.7
72.6
77.4
49.8
55.7
53.7
Natural language inference
Reading Comprehension
Open-Domain QA

Here is a goal: Get a cool sleep on summer days.
How would you accomplish this goal?
OPTIONS:
-Keep stack of pillow cases in fridge.
-Keep stack of pillow cases in oven.

keep stack of pillow cases in fridge

Translate this sentence to Spanish:
The new office building was built in less than three months.

El nuevo edificio de oficinas se construyó en tres meses.

Premise: At my age you will probably have learnt one lesson.
Hypothesis: It's not certain how many lessons you'll learn by your thirties.
Does the premise entail the hypothesis?
OPTIONS:
-yes -it is not possible to tell -no

It is not possible to tell

Figure 1: Top: overview of instruction tuning and FLAN. Instruction tuning finetunes a pretrained language model on a mixture of tasks phrased as instructions. At inference time, we evaluate on an unseen task type; for instance, we could evaluate the model on natural language inference (NLI) when no NLI tasks were seen during instruction tuning. Bottom: performance of zero-shot FLAN on unseen task types, compared with zero-shot GPT-3 and few-shot. NLI tasks: ANLI R1–R3, CB, RTE. Reading comprehension tasks: BoolQ, MultiRC, OBQA. Open-domain QA tasks: ARC-easy, ARC-challenge, NQ, TriviaQA.

[*] Lead contributors. Author contributions listed at end of paper.

1 Introduction

Language models (LMs) at scale, such as GPT-3 (Brown et al., 2020), have been shown to perform few-shot learning remarkably well. They are less successful at zero-shot learning, however. For example, GPT-3's zero-shot performance is much worse than few-shot performance on tasks such as reading comprehension, question answering, and natural language inference. One potential reason is that, without few-shot exemplars, it is harder for models to perform well on prompts that are not similar to the format of the pretraining data.

In this paper, we explore a simple method to improve the zero-shot performance of large language models, which would expand their reach to a broader audience. We leverage the intuition that NLP tasks can be described via natural language instructions, such as "*Is the sentiment of this movie review positive or negative?*" or "*Translate 'how are you' into Chinese.*" We take a pretrained language model of 137B parameters and perform *instruction tuning*—finetuning the model on a mixture of more than 60 NLP tasks expressed via natural language instructions. We refer to this resulting model as Finetuned LAnguage Net, or FLAN.

To evaluate the zero-shot performance of FLAN on unseen tasks, we group NLP tasks into clusters based on their task types and hold out each cluster for evaluation while instruction tuning FLAN on all other clusters. For example, as shown in Figure 1, to evaluate FLAN's ability to perform natural language inference, we instruction-tune the model on a range of other NLP tasks such as commonsense reasoning, translation, and sentiment analysis. As this setup ensures that FLAN has not seen any natural language inference tasks in instruction tuning, we then evaluate its ability to perform zero-shot natural language inference.

Our evaluations show that FLAN substantially improves the zero-shot performance of the base 137B-parameter model. FLAN's zero-shot also outperforms 175B-parameter GPT-3's zero-shot on 19 of 25 tasks that we evaluate, and even outperforms GPT-3's few-shot by a large margin on a number of tasks such as ANLI, RTE, BoolQ, AI2-ARC, OpenbookQA, and StoryCloze. In ablation studies, we find that increasing the number of task clusters in instruction tuning improves performance on unseen tasks and that the benefits of instruction tuning emerge only with sufficient model scale.

Our empirical results underscore the ability of language models to perform tasks described using natural language instructions. More broadly, as shown in Figure 2, instruction tuning combines appealing characteristics of the pretrain–finetune and prompting paradigms by using supervision via finetuning to improve the ability of language models to respond to inference-time text interactions. Source code for loading the instruction tuning dataset used for FLAN is made publicly available at `https://github.com/google-research/flan`.

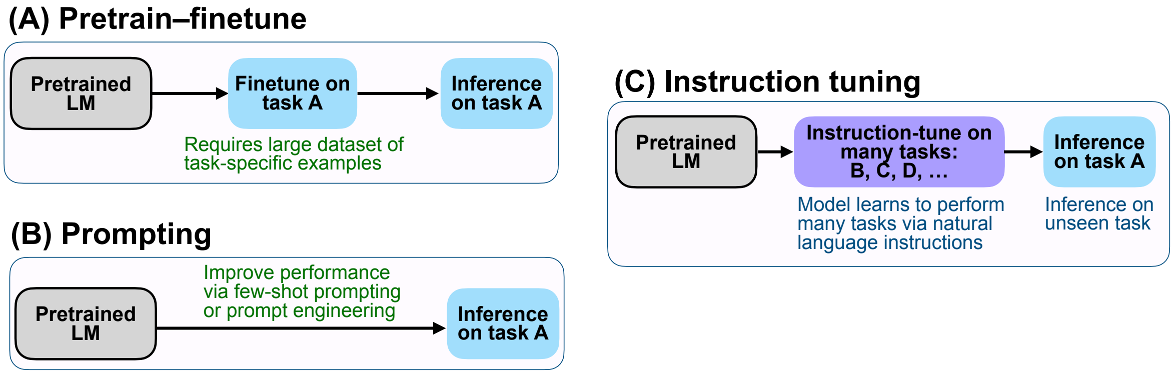

(A) Pretrain–finetune
Pretrained LM
Finetune on task A
Inference on task A
Requires large dataset of task-specific examples
(B) Prompting
Pretrained LM
Improve performance via few-shot prompting or prompt engineering
Inference on task A
(C) Instruction tuning
Pretrained LM
Instruction-tune on many tasks: B, C, D, …
Inference on task A
Model learns to perform many tasks via natural language instructions
Inference on unseen task

Figure 2: Comparing instruction tuning with pretrain–finetune and prompting.

2 FLAN: Instruction Tuning Improves Zero-Shot Learning

The motivation of instruction tuning is to improve the ability of language models to respond to NLP instructions. The idea is that by using supervision to teach an LM to perform tasks described via instructions, it will learn to follow instructions and do so even for unseen tasks. To evaluate the model's performance on unseen tasks, we group tasks into clusters by task type and hold out each task cluster for evaluation while instruction tuning on all remaining clusters.

2.1 Tasks & Templates

Creating a viable instruction tuning dataset with a large number of tasks from scratch would be resource-intensive. We therefore choose to transform existing datasets created by the research community into an instructional format. We aggregate 62 text datasets that are publicly available on Tensorflow Datasets,[1] including both language understanding and language generation tasks, into a single mixture. Figure 3 shows all datasets we use; each dataset is categorized into one of twelve task clusters, for which datasets in a given cluster are of the same task type. We refer the involved reader to Appendix B for descriptions, sizes, and examples of individual datasets.

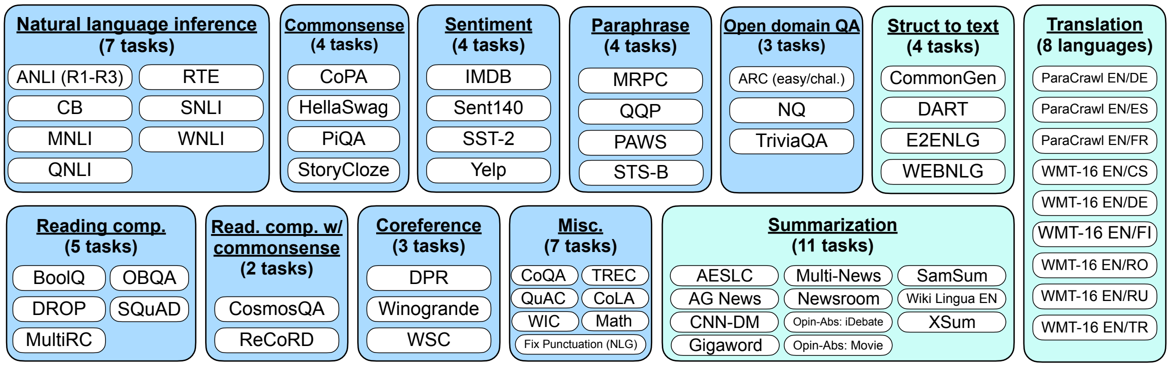


Figure 3: Task clusters used in this paper (NLU tasks in blue; NLG tasks in teal).

We define a *task* as a particular set of input–output pairs given by a dataset (e.g., we consider RTE and ANLI separate tasks, even though they are both entailment). For each task, we manually compose ten unique templates that describe the task using natural language instructions. Most of the ten templates describe the original task, but to increase diversity, for each task we include up to three templates that "turned the task around," (e.g., for sentiment classification we include templates asking to generate a negative movie review). We then instruction-tune a pretrained language model on the mixture of all tasks, with examples in each task formatted via a randomly selected instruction template for that task. Figure 4 shows multiple instruction templates for a natural language inference task.

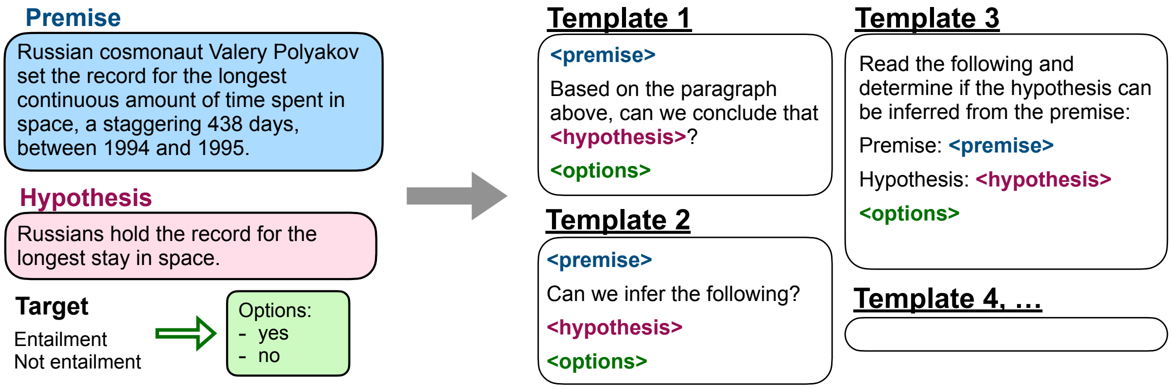

Premise
Russian cosmonaut Valery Polyakov set the record for the longest continuous amount of time spent in space, a staggering 438 days, between 1994 and 1995.
Hypothesis
Russians hold the record for the longest stay in space.
Target
Entailment
Not entailment
Options:
- yes
- no
Template 1
<premise>
Based on the paragraph above, can we conclude that <hypothesis>?
<options>
Template 2
<premise>
Can we infer the following?
<hypothesis>
<options>
Template 3
Read the following and determine if the hypothesis can be inferred from the premise:
Premise: <premise>
Hypothesis: <hypothesis>
<options>
Template 4, …

Figure 4: Multiple instruction templates describing a natural language inference task.

2.2 Evaluation Splits

We are interested in how FLAN performs on tasks that were not seen in instruction tuning, and so it is crucial to define what counts as an unseen task. Whereas some prior work has categorized unseen tasks by disallowing the same dataset to appear in training, we will use a more conservative definition that leverages the task clusters from Figure 3. In this work, we only consider task $\mathcal{T}$ unseen at evaluation time if no tasks from any clusters that

[1] https://www.tensorflow.org/datasets

$\mathcal{T}$ belongs to were seen during instruction tuning. For instance, if $\mathcal{T}$ is an entailment task, then no entailment tasks appeared in the instruction tuning dataset, and we instruction-tuned on tasks from all other clusters.[2] So, using this definition, to evaluate the performance of FLAN on tasks spanning c clusters, we instruction-tune models for c inter-cluster splits, in which each split holds out a different cluster during instruction tuning.

2.3 Classification with Options

The desired output space for a given task is either one of several given classes (e.g., classification) or free text (e.g., generation). As FLAN is an instruction-tuned version of a decoder-only language model, it naturally responds in free text, and so no further modifications are needed for tasks where the desired output is free text.

For classification tasks, prior work (Brown et al., 2020) has used a *rank classification* approach where, for example, only two outputs ("*yes*" and "*no*") are considered and the higher probability one is taken as the model's prediction. Though this procedure is logically sound, it is imperfect in that the probability mass for answers may have an undesired distribution among ways of saying each answer (e.g., a large number of alternative ways of saying "*yes*" may lower the probability mass assigned to "*yes*"). Therefore, we include an *options* suffix, in which we append the token `OPTIONS` to the end of a classification task along with a list of the output classes for that task. This makes the model aware of which choices are desired when responding to classification tasks. Example use of options is shown in the NLI and commonsense examples in Figure 1.

2.4 Training Details

Model architecture and pretraining. In our experiments, we use a dense left-to-right, decoder-only transformer language model of 137B parameters. This model is pretrained on a collection of web documents (including those with computer code), dialog data, and Wikipedia tokenized into 2.81T BPE tokens with a vocabulary of 32K tokens using the SentencePiece library (Kudo & Richardson, 2018). Approximately 10% of the pretraining data was non-English. This dataset is not as clean as the GPT-3 training set and also has a mixture of dialog and code, and so we expect the zero and few-shot performance of this pretrained LM on NLP tasks to be slightly lower. We henceforth refer to this pretrained model as *Base LM*. This same model was also previously used for program synthesis (Austin et al., 2021).

Instruction tuning procedure. FLAN is the instruction-tuned version of Base LM. Our instruction tuning pipeline mixes all datasets and randomly samples examples from each dataset. Some datasets have more than ten million training examples (e.g., translation), and so we limit the number of training examples per dataset to 30,000. Other datasets have few training examples (e.g., CommitmentBank only has 250), and so to prevent these datasets from being marginalized, we follow the examples-proportional mixing scheme (Raffel et al., 2020) with a mixing rate maximum of 3,000.[3] We finetune all models for 30,000 gradient updates at a batch size of 8,192 using the Adafactor Optimizer (Shazeer & Stern, 2018) with a learning rate of 3e-5. The input and target sequence lengths used in our finetuning procedure are 1024 and 256 respectively. We use packing (Raffel et al., 2020) to combine multiple training examples into a single sequence, separating inputs from targets using a special end-of-sequence token.

3 Results

In this section, we evaluate FLAN on tasks spanning natural language inference, reading comprehension, open-domain QA, commonsense reasoning, coreference resolution, and translation. As described in Section 2.2, we evaluate on unseen tasks by grouping tasks into clusters and holding out each cluster for evaluation while instruction tuning on all remaining clusters. For each task we report the mean and standard deviation of performance on all templates, which proxies the expected performance of FLAN given a typical natural language instruction. As a dev set is sometimes available for manual prompt engineering (Brown et al., 2020), for each task we also report the test set performance using the template that had the best dev set performance.

[2] When evaluating on the reading comprehension with commonsense cluster, both reading comprehension and commonsense reasoning were dropped from instruction tuning. Conversely, the reading comprehension with commonsense cluster was not used for instruction tuning when evaluating on reading comprehension or commonsense reasoning. We also drop the paraphrase cluster from instruction tuning when evaluating on NLI tasks and vice-versa.

[3] In this mixing scheme, a mixing rate maximum of 3,000 means that a dataset does not receive additional sampling weight for examples in excess of 3,000. In other words, a dataset i with n_i examples would have sampling frequency $w_i = \min(n_i, 3000)/\sum_{j\in\mathcal{D}}\min(n_j, 3000)$, where $\mathcal{D}$ is the set of all datasets.

For comparison, we include the zero-shot/few-shot performance of 175B GPT-3. With the best dev template, zero-shot FLAN outperforms zero-shot GPT-3 on 19 out of 25 tasks and even surpasses GPT-3's few-shot performance on 10 tasks. We also report zero and few-shot[4] results from Base LM using the same prompts as GPT-3, as Base LM is not suitable for natural instructions without instruction tuning. This baseline provides the most direct ablation of how much instruction tuning helps. Instruction tuning significantly improves Base LM on most tasks. In terms of supervised models, we report the performance of T5-11B (Raffel et al., 2020) when available; otherwise, we show the performance of BERT-large (Devlin et al., 2019).

Overall, we observe that instruction tuning is very effective on tasks that can be naturally verbalized as instructions (e.g., natural language inference and question answering) and is less effective on tasks that are directly formulated as language modeling (e.g., commonsense reasoning and coreference resolution).

3.1 Natural Language Inference

Table 1 shows results for natural language inference (NLI), where given a premise and hypothesis, a model must determine whether the hypothesis is true given that the premise is true. FLAN has strong performance in all cases. Despite high variance among results for different templates for CB (de Marneffe et al., 2019) and RTE (Wang et al., 2018), FLAN outperforms zero and few-shot GPT-3 for four datasets by a large margin without any prompt engineering. With the best dev template, FLAN outperforms few-shot GPT-3 on all five datasets. FLAN even surpasses supervised BERT on ANLI-R3 (Nie et al., 2020).

As noted by Brown et al. (2020), perhaps one reason why GPT-3 struggles with NLI is that NLI examples are unlikely to have appeared naturally in an unsupervised training set and are thus awkwardly phrased as a continuation of a sentence. For FLAN on the other hand, we phrase NLI as the more natural question "`Does <premise> mean that <hypothesis>?`" and achieve much higher performance.

	Natural Language Inference				
	ANLI-R1 acc.	ANLI-R2 acc.	ANLI-R3 acc.	CB acc.	RTE acc.
Supervised model	57.4[b]	48.3[b]	43.5[b]	96.8[a]	92.5[a]
Base LM 137B zero-shot	39.6	39.9	39.3	42.9	73.3
· few-shot	39.0	37.5	40.7	34.8	70.8
GPT-3 175B zero-shot	34.6	35.4	34.5	46.4	58.9
· few-shot	36.8	34.0	40.2	82.1	70.4
FLAN 137B zero-shot					
- no prompt engineering	47.7 ▲10.9 stdev=1.4	43.9 ▲8.5 stdev=1.3	47.0 ▲6.8 stdev=1.4	64.1 ↑17.7 stdev=14.7	78.3 ▲7.9 stdev=7.9
- best dev template	46.4 ▲9.6	44.4 ▲9.0	48.5 ▲8.3	83.9 ▲1.8	84.1 ▲13.9

Table 1: Results on natural language inference. For FLAN, we report both the average of up to ten templates (proxying the expected performance without prompt engineering), as well as the test set performance of the template that had the highest performance on the dev set. The triangle ▲ indicates improvement over few-shot GPT-3. The up-arrow ↑ indicates improvement only over zero-shot GPT-3. [a]T5-11B, [b]BERT-large.

3.2 Reading Comprehension & Open-Domain QA

Results for reading comprehension, where models are asked to answer a question about a provided passage, are shown in Table 2. FLAN outperforms GPT-3 by a large margin on BoolQ (Clark et al., 2019a) and OBQA (Mihaylov et al., 2018), although Base LM already achieves high performance on BoolQ. On MultiRC (Khashabi et al., 2018), the performance of FLAN is slightly higher than few-shot GPT-3 when using the best dev template.

Also shown in Table 2 are results for open-domain question answering, which asks models to answer questions about the world without access to specific information containing the answer. On ARC-easy and ARC-challenge (Clark et al., 2018), the performance of FLAN is substantially higher than both zero-shot and few-shot GPT-3. For Natural Questions (NQ; Lee et al., 2019; Kwiatkowski et al., 2019) FLAN outperforms zero-shot GPT-3 but does not outperform few-shot GPT-3. For TriviaQA (Joshi et al., 2017), however, FLAN underperforms zero-shot GPT-3, which achieves remarkable performance and beats supervised T5-11B.

[4] Unlike GPT-3, we do not choose the number of few-shot examples by searching over a dev set. Instead, we either used ten-shot results or the maximum number of exemplars when ten exemplars did not fit into our context length of 1024 tokens.

	READING COMPREHENSION			OPEN-DOMAIN QA			
	BoolQ acc.	MultiRC F1	OBQA acc.	ARC-e acc.	ARC-c acc.	NQ EM	TriviaQA EM
Supervised model	91.2[a]	88.2[a]	85.4[a]	92.6[a]	81.1[a]	36.6[a]	60.5[a]
Base LM 137B zero-shot	81.0	60.0	41.8	76.4	42.0	3.2	18.4
· few-shot	79.7	59.6	50.6	80.9	49.4	22.1	55.1
GPT-3 175B zero-shot	60.5	72.9	57.6	68.8	51.4	14.6	64.3
· few-shot	77.5	74.8	65.4	70.1	51.5	29.9	71.2
FLAN 137B zero-shot							
- no prompt engineering	80.2 ▲2.7 stdev=3.1	74.5 ↑2.4 stdev=3.7	77.4 ▲12.0 stdev=1.3	79.5 ▲8.6 stdev=0.8	61.7 ▲10.2 stdev=1.4	18.6 ▲4.0 stdev=2.7	55.0 stdev=2.3
- best dev template	82.9 ▲5.4	77.5 ▲2.7	78.4 ▲13.0	79.6 ▲8.7	63.1 ▲11.6	20.7 ▲6.1	56.7

Table 2: Results on reading comprehension and open-domain question answering. For FLAN, we report both the average of up to ten templates (proxying the expected performance without prompt engineering), as well as the test set performance of the template that had the highest performance on the dev set. The triangle ▲ indicates improvement over few-shot GPT-3. The up-arrow ↑ indicates improvement only over zero-shot GPT-3. [a]T5-11B.

3.3 COMMONSENSE REASONING & COREFERENCE RESOLUTION

Results for five commonsense reasoning datasets are shown in Table 3. FLAN outperforms GPT-3 on StoryCloze (Mostafazadeh et al., 2016) and has comparable performance on CoPA (Roemmele et al., 2011) and PiQA (Bisk et al., 2020). For HellaSwag (Zellers et al., 2019) and ReCoRD (Zhang et al., 2018), however, both Base LM and FLAN performs well below GPT-3.

Table 3 also shows results for two coreference resolution datasets. FLAN with the best dev template outperforms zero-shot GPT-3 on Winogrande (Sakaguchi et al., 2020). For WSC273 (Levesque et al., 2012), both Base LM and FLAN are well below GPT-3 (we cannot compute the best dev template result for FLAN since WSC273 does not have a validation set).

One observation is that HellaSwag, PiQA, ReCoRD, WSC273, and Winogrande are directly formulated in the original pretraining objective of language modeling. Hence, the result that GPT-3 outperforms FLAN by a large margin is consistent with the intuition that gains from FLAN will be marginal when instructions are not crucial for describing the given task. Moreover, we note a further limitation with FLAN for these five language modeling tasks, including options actually hurts performance, and so the reported results are for rank classification without options.

	COMMONSENSE REASONING					COREFERENCE	
	CoPA acc.	HellaSwag acc.	PiQA acc.	StoryCloze acc.	ReCoRD acc.	WSC273 acc.	Winogrande acc.
Supervised model	94.8[a]	47.3[b]	66.8[b]	89.2[b]	93.4[a]	72.2[b]	93.8[a]
Base LM 137B zero-shot	90.0	57.0	80.3	79.5	87.8	81.0	68.3
· few-shot	89.0	58.8	80.2	83.7	87.6	61.5	68.4
GPT-3 175B zero-shot	91.0	78.9	81.0	83.2	90.2	88.3	70.2
· few-shot	92.0	79.3	82.3	87.7	89.0	88.6	77.7
FLAN 137B zero-shot							
- no prompt engineering	90.6 stdev=2.0	56.4 stdev=0.5	80.9 stdev=0.8	92.2 ▲4.5 stdev=1.3	67.8 stdev=3.0	80.8 stdev=3.7	67.3 stdev=2.5
- best dev template	91.0	56.7	80.5	93.4 ▲5.7	72.5	-	71.2 ↑1.0

Table 3: Results (accuracy in %) for commonsense reasoning and coreference resolution. For FLAN, we report both the average of up to ten templates (proxying the expected performance without prompt engineering), as well as the test set performance of the template that had the highest performance on the dev set. [a]T5-11B, [b]BERT-large. The triangle ▲ indicates improvement over few-shot GPT-3. The up-arrow ↑ indicates improvement only over zero-shot GPT-3.

3.4 Translation

Similar to GPT-3, the training data for Base LM is around 90% English and includes some text in other languages that was not specifically used to train the model to perform machine translation. We also evaluate FLAN's performance on machine translation for the three datasets evaluated in the GPT-3 paper: French–English from WMT'14 (Bojar et al., 2014), and German–English and Romanian–English from WMT'16 (Bojar et al., 2016).

These results are shown in Table 4. For Base LM, zero-shot translation had low performance, but few-shot translation results were comparable with GPT-3; FLAN outperforms few-shot Base LM in five of the six evaluations. Compared with GPT-3, FLAN outperforms zero-shot GPT-3 for all six evaluations, though it underperforms few-shot GPT-3 in most cases. Similar to GPT-3, FLAN shows strong results for translating into English and compares favorably against supervised translation baselines. Translating from English into other languages, however, was relatively weaker, as might be expected given that FLAN uses an English sentencepiece tokenizer and that the majority of pretraining data is English.

	Translation					
	French		German		Romanian	
	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
	BLEU	BLEU	BLEU	BLEU	BLEU	BLEU
Supervised model	45.6^c	35.0^d	41.2^e	38.6^f	38.5^g	39.9^g
Base LM 137B zero-shot	11.2	7.2	7.7	20.8	3.5	9.7
· few-shot	31.5	34.7	26.7	36.8	22.9	37.5
GPT-3 175B zero-shot	25.2	21.2	24.6	27.2	14.1	19.9
· few-shot	32.6	39.2	29.7	40.6	21.0	39.5
FLAN 137B zero-shot						
- no prompt engineering	32.0 ↑6.8 stdev=2.0	35.6 ↑14.4 stdev=1.5	24.2 stdev=2.7	39.4 ↑12.2 stdev=0.6	16.9 ↑2.8 stdev=1.4	36.1 ↑16.2 stdev=1.0
- best dev template	34.0 ▲1.4	36.5 ↑15.3	27.0 ↑2.4	39.8 ↑12.6	18.4 ↑4.3	36.7 ↑16.7

Table 4: Translation results (BLEU) for WMT'14 En/Fr and WMT'16 En/De and En/Ro. For FLAN, we report both the average of up to ten templates (proxying the expected performance without prompt engineering), as well as the test set performance of the template that had the highest performance on the dev set. [c]Edunov et al. (2018), [d]Durrani et al. (2014), [e]Wang et al. (2019b), [f]Sennrich et al. (2016), [g]Liu et al. (2020). The triangle ▲ indicates improvement over few-shot GPT-3. The up-arrow ↑ indicates improvement only over zero-shot GPT-3.

4 Ablation Studies & Further Analysis

4.1 Number of instruction tuning clusters

As the core question of our paper asks how instruction tuning improves a model's zero-shot performance on unseen tasks, in this first ablation we examine how performance is affected by the number of clusters and tasks used in instruction tuning. For this setup, we hold out NLI, open-domain QA, and commonsense reasoning as evaluation clusters, and use the seven remaining clusters for instruction tuning.[5] We show results for 1-7 instruction tuning clusters, where clusters are added in decreasing order of number of tasks per cluster.

Figure 5 shows these results. As expected, we observe that average performance across the three held-out clusters improves as we add additional clusters and tasks to instruction tuning (with the exception of the sentiment analysis cluster), confirming the benefits of our proposed instruction tuning approach on zero-shot performance on novel tasks. It is further interesting to see that, for the seven clusters we test, the performance does not appear to saturate, implying that performance may further improve with even more clusters added to instruction tuning. Of note, this ablation does not allow us to draw conclusions about which instruction tuning cluster contributes the most to each evaluation cluster, although we see minimal added value from the sentiment analysis cluster.

[5] We do not use the paraphrase or reading comprehension with commonsense clusters for instruction tuning in this ablation because they are too similar to NLI and commmonsense reasoning, respectively.

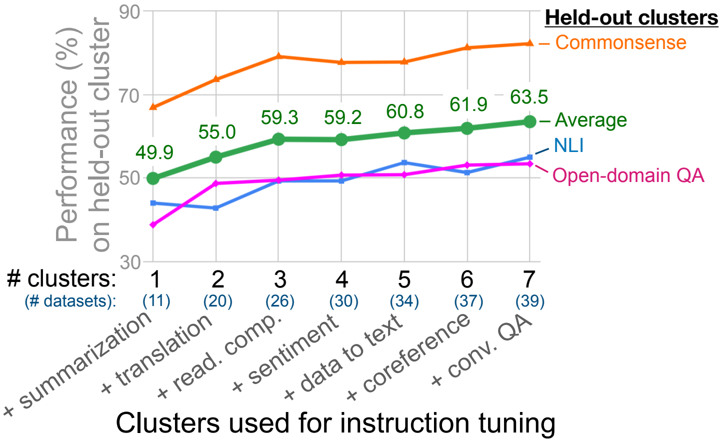


Figure 5: Adding additional task clusters to instruction tuning improves zero-shot performance on held-out task clusters. The evaluation tasks are the following. Commonsense: CoPA, HellaSwag, PiQA, and StoryCloze. NLI: ANLI R1–R3, QNLI, RTE, SNLI, and WNLI. Open-domain QA: ARC easy, ARC challenge, Natural Questions, and TriviaQA.

4.2 Scaling laws

As Brown et al. (2020) shows that zero and few-shot capabilities of language models substantially improve for larger models, we next explore how the benefits of instruction tuning are affected by model scale. Using the same cluster split as in the previous ablation study, we evaluate the effect of instruction tuning on models of size 422M, 2B, 8B, 68B, and 137B parameters.

Figure 6 shows these results. First, as a check of correctness, we confirm in Figure 6A that, for all model sizes, performance on tasks that were seen during instruction tuning improves by a large margin over the untuned model without instruction tuning, as would be highly expected. As for held-out tasks, in Figure 6B we see that for the two models on the order of 100B parameters, instruction tuning substantially improves performance, as is also expected given the prior results in our paper. The behavior on held-out tasks for the 8B and smaller models, however, is thought-provoking—instruction tuning actually hurts performance on held-out tasks. One potential explanation for this result could be that for small-scale models, learning the ∼40 tasks used during instruction tuning fills the entire model capacity, causing these models to perform worse on new tasks. For the larger scale models, instruction tuning fills up some model capacity but also teaches these models the ability to follow instructions, allowing them to generalize to new tasks with the remaining capacity.

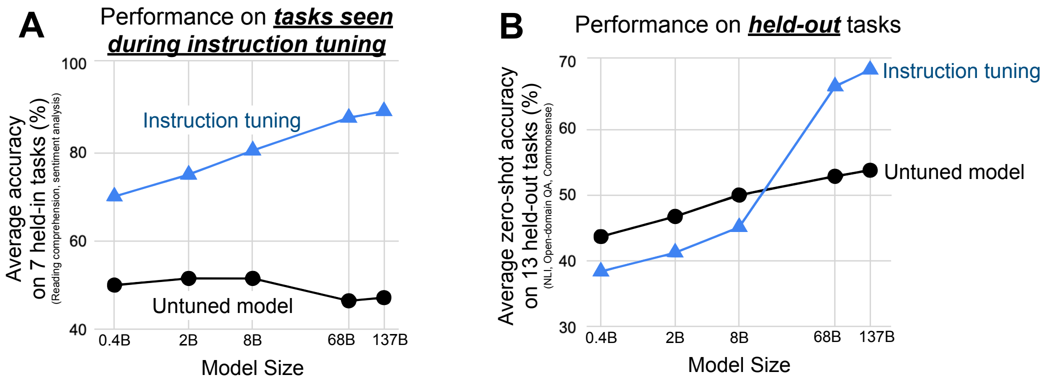


Figure 6: (A) Performance on tasks seen during instruction tuning improves for all model sizes. (B) The effect of instruction tuning on performance on unseen tasks depends on the model scale. Whereas instruction tuning helps large models generalize to new tasks, for small models it actually hurts generalization to unseen tasks, potentially because all model capacity is used to learn the mixture of instruction tuning tasks.

4.3 Instruction Tuning Facilitates Prompt Tuning

As we've seen that instruction tuning improves the ability of a model to respond to instructions, it follows that, if FLAN is indeed more amenable to performing NLP tasks, then it should also achieve better performance when performing inference using continuous prompts optimized via prompt tuning (Li & Liang, 2021; Lester et al., 2021). As further analysis, we train continuous prompts for each of the SuperGLUE (Wang et al., 2019a) tasks in accordance with the cluster splits from Section 2.2 such that when prompt-tuning on task $\mathcal{T}$, no tasks in the same cluster as $\mathcal{T}$ were seen during instruction tuning. Our prompt tuning setup follows the procedure of Lester et al. (2021) except that we use a prompt length of 10, weight decay of 1e-4, and did not use dropout on the attention scores; we found in preliminary experiments that these changes improved the performance of Base LM.

Table 5 shows the results of these prompt tuning experiments for both using a fully-supervised training set and in a low-resource setting with only 32 training examples. We see that in all scenarios, prompt tuning works better with FLAN than Base LM. In many cases, especially for the low-resource setting, prompt tuning on FLAN even achieves more than 10% improvement over prompt tuning on the Base LM. This result exemplifies in another way how instruction tuning can result in a checkpoint that is more desirable for performing NLP tasks.

		Prompt Tuning Analysis							
	Prompt tuning train. examples	BoolQ acc.	CB acc.	CoPA acc.	MultiRC F1	ReCoRD acc.	RTE acc.	WiC acc.	WSC acc.
Base LM	32	55.5	55.4	87.0	65.4	78.0	52.4	51.6	65.4
FLAN		77.5	87.5	91.0	76.8	80.8	83.0	57.8	70.2
Base LM	full dataset	82.8	87.5	90.0	78.6	84.8	82.0	54.9	72.7
FLAN		86.3	98.2	94.0	83.4	85.1	91.7	74.0	86.5

Table 5: FLAN responds better to continuous inputs attained via prompt tuning than Base LM. When prompt tuning on a given dataset, no tasks from the same cluster as that dataset were seen during instruction tuning.

5 Related Work

Language models and multi-task learning. Our work is broadly inspired by a long line of prior work on language models for NLP applications (Dai & Le, 2015; Peters et al., 2018; Howard & Ruder, 2018; Radford et al., 2018; 2019, *inter alia*). Instruction tuning can be seen as a formulation of multitask learning (MTL), which is an established area within deep learning (Collobert et al., 2011; Luong et al., 2016; Ruder, 2017; Velay & Daniel, 2018; Clark et al., 2019b; Liu et al., 2019b, *inter alia*)—see Worsham & Kalita (2020) for a recent survey on MTL for NLP. Differing from prior MTL work which focuses on performance improvements across training tasks (Raffel et al., 2020; Aghajanyan et al., 2021) or to new domains (Axelrod et al., 2011), our work focuses on zero-shot generalization to tasks that were not seen in training.

Zero-shot learning. Our work also falls in the well-established category of zero-shot learning, which has historically been used to refer to classifying instances among a set of unseen categories (Lampert et al., 2009; Romera-Paredes & Torr, 2015; Srivastava et al., 2018; Yin et al., 2019, *inter alia*). In NLP, zero-shot learning work also includes translating between unseen language pairs (Johnson et al., 2017; Pham et al., 2019), language modeling on unseen languages (Lauscher et al., 2020), as well as various NLP applications (Liu et al., 2019a; Corazza et al., 2020; Wang et al., 2021; Reif et al., 2021). Most recently, the emergent ability of language models (Brown et al., 2020) has led to increased interest in how models generalize to unseen tasks, the definition of zero-shot learning used in our paper. In addition, meta-learning (Finn et al., 2017; Vanschoren, 2018, *inter alia*) also broadly tries to train models that adapt quickly to unseen tasks.

Prompting. Instruction tuning leverages the intuition that language models at scale contain substantial world knowledge and can perform a range of NLP tasks (Brown et al., 2020, see also Bommasani et al. (2021)). Another line of work that shares this goal prompts models with continuous inputs optimized via backpropagation to substantially improve performance (Li & Liang, 2021; Lester et al., 2021; Qin & Eisner, 2021). Although the success of these approaches depends heavily on model scale (Lester et al., 2021), for which large models can be costly to serve, the ability of a single large model to perform many tasks slightly eases this burden.

Multi-task question answering. The instructions we use for instruction tuning are similar to multi-task question answering work such as McCann et al. (2018), which casts ten NLP tasks as question answering over a context and trains a model on a collection of tasks formulated with natural language prompts. They report transfer learning gains on finetuning tasks as well as zero-shot domain adaptation results on SNLI (Bowman et al., 2015) and Amazon/Yelp Reviews (Kotzias et al., 2015). While McCann et al. (2018) does not leverage unsupervised pre-training and only reports zero-shot transfer to unseen domains, our work uses a pretrained LM and focuses on zero-shot performance on unseen task clusters. Focusing on binary text classification, Zhong et al. (2021) finetune T5-770M on 43 tasks phrased as yes/no questions and study the zero-shot performance on unseen tasks. In comparison, our paper is much larger in scope, empirically demonstrating the idea on a wide range of tasks with a much larger model.

Instructions-based NLP. The recent improvements in the capabilities of language models have led to increased interest in a nascent area of instructions-based NLP (Goldwasser & Roth, 2014, and see McCarthy (1960)). Schick & Schütze (2021) use task descriptions in cloze-style phrases to help language models assign soft labels for semi-supervised learning. Efrat & Levy (2020) evaluated GPT-2 (Radford et al., 2019) on simple tasks ranging from retrieving the nth word of a sentence to generating examples for SQuAD, concluding that GPT-2 performs poorly across all tasks. Mishra et al. (2021) finetune BART (Lewis et al., 2020) using instructions for tasks such as question answering, text classification, and text modification. In addition to differing in model scale and number of evaluation tasks, our work crucially differs from theirs in that we study generalization to novel tasks. Whereas Mishra et al. (2021) finetune and evaluate on task distributions that are similar, the inter-cluster splits we use in this paper evaluate the ability for models to perform new tasks when no similar tasks have been seen during finetuning. Ye et al. (2021) introduce a setup for cross-task few-shot learning, finding that multi-task learning improves the few-shot capabilities of BART on an unseen downstream task. Finally, OpenAI has a beta product called the Instruct Series. Though they have not released a research paper on it at this time, based on examples posted on their website,[6] their product appears to have goals similar to those in our work.

6 Discussion

Our paper has explored a simple question in zero-shot prompting: does instruction tuning a language model improve its ability to perform unseen tasks? Our experiments on FLAN demonstrate that instruction tuning improves performance against an untuned model and surpasses zero-shot GPT-3 on the majority of tasks that we evaluate on. Through ablation studies, we learn that performance on unseen tasks improves with the number of task clusters used in instruction tuning, and, interestingly, that the benefits of instruction tuning emerge only with sufficient model scale. Moreover, our FLAN appears to respond better to prompt tuning than the unmodified base model, demonstrating an additional benefit of instruction tuning.

One limitation in our study is that there is a certain degree of subjectivity in grouping tasks into clusters (for instance, sentiment analysis can be seen as a small subset of reading comprehension), as there is no accepted method for operationalizing the similarity between two tasks. Hence, we assigned tasks to clusters based on accepted categorizations in the literature, taking the conservative approach when tasks seemed to fall into multiple clusters (such as excluding reading comprehension with commonsense from instruction tuning when evaluating both reading comprehension and commonsense reasoning). As another limitation, we used short instructions (typically a single sentence) to describe well-known NLP tasks. Other tasks may require much longer or more specific instructions to describe adequately, along with explanations involving examples; we leave these scenarios for future work.

The results shown in this paper suggest several promising directions for future research. Although FLAN is instruction-tuned on over sixty datasets, these datasets only cover ten task clusters (plus some miscellaneous tasks), a relatively small number considering all potential tasks such a model could be used for. It is possible that performance can further improve with many more instruction tuning tasks, which could be generated in a self-supervised fashion, for instance. In addition to gathering more tasks, it will also be valuable to explore the multilingual setting, where one could ask, for instance, whether instruction tuning on supervised data in high-resource languages would improve performance on novel tasks in low-resource languages. Finally, instruction tuning models with supervised data could also potentially be used to improve model behavior with respect to bias and fairness (Solaiman & Dennison, 2021).

[6] https://beta.openai.com/docs/engines/instruct-series-beta

7 Conclusions

This paper has explored instruction tuning. We presented FLAN, a 137B parameter language model that performs NLP tasks described using instructions. By leveraging supervision from finetuning to improve the ability of language models to respond to instructional prompts, FLAN combines appealing aspects of both the pretrain–finetune and prompting paradigms. The performance of FLAN compares favorably against both zero-shot and few-shot GPT-3, signaling the potential ability for models at scale to follow instructions. We hope that our paper will spur further research on zero-shot learning and using labeled data to improve language models.

Ethical Considerations

This work uses language models, for which the risks and potential harms are discussed in Bender & Koller (2020), Brown et al. (2020), Bender et al. (2021), Patterson et al., (2021), and others. As our contribution in this paper is not a pretrained language model itself but rather an empirical study of how instruction tuning affects the zero-shot performance of a language model on unseen tasks, we additionally highlight several relevant ethical considerations. First, labeled datasets such as those we use for finetuning can contain undesirable biases, and these biases can be propagated into zero-shot applications of the model on downstream tasks. Next, instruction-tuned models can potentially require less data and expertise to use; such lower barriers to access could increase both the benefits and associated risks of such models. Finally, we want to be clear that, for most datasets, supervised models such as BERT and T5 still outperform the zero-shot instructions-only model in this paper.

Environmental Considerations

We use the same pretrained language models as Austin et al. (2021). The energy cost and carbon footprint for those pretrained models were 451 MWh and 26 tCO2e, respectively. The additional instruction tuning gradient-steps for finetuning flan is less than 2% of the number of pretraining steps, and so the estimated additional energy cost is comparatively smaller.

Author Contributions

Maarten Bosma conceived the original idea and implemented the first version of FLAN. Vincent Zhao prototyped the training and evaluation pipelines, as well as rank classification. Kelvin Guu proposed and implemented the idea of task clusters and evaluation using inter-cluster splits. Jason Wei, Maarten Bosma, Vincent Zhao, and Adams Wei Yu implemented the NLP tasks. Jason Wei, Vincent Zhao, and Adams Wei Yu conducted and managed the experiments. Jason Wei designed and ran the ablation studies. Jason Wei, Maarten Bosma, and Quoc V. Le wrote most of the paper. Jason Wei, Maarten Bosma, and Nan Du obtained the zero and few-shot baselines. Brian Lester ran the prompt tuning experiments. Quoc V. Le and Andrew M. Dai advised, provided high-level guidance and helped edit the paper.

Acknowledgements

We thank Ed Chi, Slav Petrov, Dan Garrette, and Ruibo Liu for providing feedback on our manuscript. We thank Adam Roberts, Liam Fedus, Hyung Won Chung, and Noam Shazeer for helping debug some of our models. We thank Ellie Pavlick for feedback on the study design during the middle stages of the project. We thank Daniel De Freitas Adiwardana for helping initiate the project, large language model advising, and giving us access to some computational resources. Finally, we thank the team involved in pretraining of the Base LM model: De Freitas Adiwardana, Noam Shazeer, Yanping Huang, Dmitry Lepikhin, Dehao Chen, Yuanzhong Xu and Zhifeng Chen.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning. 2021. URL https://arxiv.org/abs/2101.11038.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. 2021. URL https://arxiv.org/abs/2108.07732.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 355–362, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1033.

Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5185–5198, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.463. URL https://aclanthology.org/2020.acl-main.463.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pp. 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *TAC*, 2009. URL https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.232.1231&rep=rep1&type=pdf.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020. URL https://arxiv.org/abs/1911.11641.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, and Lucia Specia (eds.). *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-33. URL https://aclanthology.org/W14-3300.

Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Liane Guillou, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Pavel Pecina, Martin Popel, Philipp Koehn, Christof Monz, Matteo Negri, Matt Post, Lucia Specia, Karin Verspoor, Jörg Tiedemann, and Marco Turchi (eds.). *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2200. URL https://aclanthology.org/W16-2200.

Rishi Bommasani, Drew A. Hudson, E. Adeli, R. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, E. Brynjolfsson, S. Buch, D. Card, Rodrigo Castellon, Niladri S. Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, S. Ermon, J. Etchemendy, Kawin Ethayarajh, L. Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, S. Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, G. Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, M. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, J. Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, D. Narayanan, Ben Newman, Allen Nie, J. C. Niebles, H. Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, C. Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, D. Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, K. Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, M. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. 2021.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 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)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5931–5937, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1595. URL https://aclanthology.org/P19-1595.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. 2018. URL https://arxiv.org/abs/1803.05457.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011. URL https://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf.

Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. Hybrid emoji-based masked language models for zero-shot abusive language detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 943–949, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.84. URL https://aclanthology.org/2020.findings-emnlp.84.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, pp. 177–190, 2005. doi: 10.1007/11736790_9. URL https://doi.org/10.1007/11736790_9.

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Proceedings of the Conference on Neural Information Processing Systems*, 2015. URL https://papers.nips.cc/paper/2015/file/7137debd45ae4d0ab9aa953017286b20-Paper.pdf.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. The CommitmentBank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung*, 23(2):107–124, Jul. 2019. doi: 10.18148/sub/2019.v23i2.601. URL https://ojs.ub.uni-konstanz.de/sub/index.php/sub/article/view/601.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. Edinburgh's phrase-based machine translation systems for WMT-14. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 97–104, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3309. URL `https://aclanthology.org/W14-3309`.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1045. URL `https://aclanthology.org/D18-1045`.

Avia Efrat and Omer Levy. The Turking Test: Can language models understand instructions? 2020. URL `https://arxiv.org/abs/2010.11982`.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1126–1135. PMLR, 2017. URL `https://arxiv.org/abs/1703.03400`.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 1–9, Prague, June 2007. Association for Computational Linguistics. URL `https://aclanthology.org/W07-1401`.

Dan Goldwasser and Dan Roth. Learning from natural instructions. *Machine learning*, 94(2):205–232, 2014. URL `https://link.springer.com/article/10.1007/s10994-013-5407-y`.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The Second PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006. URL `http://www.cs.biu.ac.il/~szpekti/papers/RTE2-organizers.pdf`.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL `https://aclanthology.org/P18-1031`.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065. URL `https://aclanthology.org/Q17-1024`.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 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)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL `https://aclanthology.org/P17-1147`.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 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)*, pp. 252–262, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1023. URL `https://aclanthology.org/N18-1023`.

Dimitrios Kotzias, Misha Denil, Nando de Freitas, and Padhraic Smyth. From group to individual labels using deep features. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015. URL `https://dl.acm.org/doi/10.1145/2783258.2783380`.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pp. 66–71. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-2012. URL `https://doi.org/10.18653/v1/d18-2012`.

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. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, March 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 951–958. IEEE, 2009.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4483–4499, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.363. URL https://aclanthology.org/2020.emnlp-main.363.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1612. URL https://aclanthology.org/P19-1612.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://arxiv.org/abs/2104.08691.

Hector Levesque, Ernest Davis, and Leora Morgenstern. The Winograd Schema Challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012. URL https://dl.acm.org/doi/10.5555/3031843.3031909.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Xiang Lisa Li and Percy Liang. 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)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353.

Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert Y.S. Lam. Reconstructing capsule networks for zero-shot intent classification. 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)*, pp. 4799–4809, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1486. URL https://aclanthology.org/D19-1486.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4487–4496, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1441. URL https://aclanthology.org/P19-1441.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020. doi: 10.1162/tacl_a_00343. URL https://aclanthology.org/2020.tacl-1.47.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *Proceedings of ICLR*, 2016. URL https://arxiv.org/abs/1511.06114.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. 2018. URL https://arxiv.org/abs/1806.08730.

John McCarthy. *Programs with common sense*. RLE and MIT computation center, 1960. URL http://jmc.stanford.edu/articles/mcc59/mcc59.pdf.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 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*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Natural Instructions: Benchmarking generalization to new tasks from natural language instructions. *arXiv preprint arXiv:2104.08773*, 2021. URL https://arxiv.org/abs/2104.08773.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 839–849, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1098. URL https://aclanthology.org/N16-1098.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.441. URL https://aclanthology.org/2020.acl-main.441.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

Ngoc-Quan Pham, Jan Niehues, Thanh-Le Ha, and Alexander Waibel. Improving zero-shot translation with language-independent constraints. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pp. 13–23, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5202. URL https://aclanthology.org/W19-5202.

Guanghui Qin and Jason Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 5203–5212, Online, June 2021. URL http://cs.jhu.edu/~jason/papers/#qin-eisner-2021.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. https://blog.openai.com/language-unsupervised, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. A recipe for arbitrary text style transfer with large language models. 2021.

Melissa Roemmele, Cosmin Bejan, and Andrew Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning, 2011. URL https://www.aaai.org/ocs/index.php/SSS/SSS11/paper/view/2418.

Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *International conference on machine learning*, pp. 2152–2161. PMLR, 2015.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. 2017. URL https://arxiv.org/abs/1706.05098.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. URL https://arxiv.org/abs/1907.10641.

Timo Schick and Hinrich Schütze. 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*, pp. 255–269, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.20.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pp. 371–376, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2323. URL https://aclanthology.org/W16-2323.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pp. 4596–4604. PMLR, 2018. URL https://arxiv.org/abs/1804.04235.

Irene Solaiman and Christy Dennison. Process for adapting language models to society (palms) with values-targeted datasets. *arXiv preprint arXiv:2106.10328*, 2021. URL https://cdn.openai.com/palms.pdf.

Shashank Srivastava, Igor Labutov, and Tom Mitchell. Zero-shot learning of classifiers from natural language quantification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 306–316, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1029. URL https://aclanthology.org/P18-1029.

Joaquin Vanschoren. Meta-learning: A survey. 2018. URL https://arxiv.org/abs/1810.03548.

Marc Velay and Fabrice Daniel. Seq2seq and multi-task learning for joint intent and content extraction for domain specific interpreters. 2018. URL https://arxiv.org/abs/1808.00423.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 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*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*, 2019a.

Yiren Wang, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. Multi-agent dual learning. In *Proceedings of the International Conference on Learning Representations (ICLR) 2019*, 2019b. URL https://openreview.net/forum?id=HyGhN2A5tm.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. Towards zero-label language learning. 2021.

Joseph Worsham and J. Kalita. Multi-task learning for natural language processing in the 2020s: where are we going? 2020. URL https://arxiv.org/abs/2007.16008.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. Crossfit: A few-shot learning challenge for cross-task generalization in nlp. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://arxiv.org/abs/2104.08835.

Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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)*, pp. 3914–3923, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1404. URL https://aclanthology.org/D19-1404.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL `https://aclanthology.org/P19-1472`.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. Record: Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, abs/1810.12885, 2018. URL `http://arxiv.org/abs/1810.12885`.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. Meta-tuning language models to answer prompts better. *arXiv preprint arXiv:2104.04670*, 2021. URL `https://arxiv.org/abs/2104.04670`.

A Further Ablation Studies

A.1 Tasks per Cluster & Templates per Task

Our primary hypothesis is that that instruction tuning on a diverse set of tasks improves performance on unseen tasks. Section 4.1 showed that adding more clusters improves performance; here, we further explore whether adding additional tasks improves performance when the number of clusters is held constant. We use the same split as in Section 4.1, where the NLI, commonsense reasoning, and open-domain QA clusters are held-out, and seven other clusters remain for instruction tuning. For these seven clusters, we instruction-tune models using just one task per cluster and using four tasks per cluster (for clusters that did not have four tasks, we just used all available tasks). In addition, we simultaneously explore the role of the number of instruction templates per task; as mentioned in Section 2.1, for each task we manually composed ten instructional templates for instruction tuning. Here, we instruction-tune models using 1, 4, and 10 templates per task.

Section 7 shows these results. Using more tasks per cluster improved performance by almost 10% on average across the three held-out clusters. Using more templates per task, however, had a comparatively negligible effect on performance when there was one task per cluster, which disappeared when there were four tasks per cluster. The small effect of templates is striking given our original motivation that composing ten templates per task would mitigate overfitting to any particular template. This results serves to underscore, however, the unpredictability of finetuning large language models, as one hypothesis is that models at such scale do not easily overfit to a single task during finetuning.

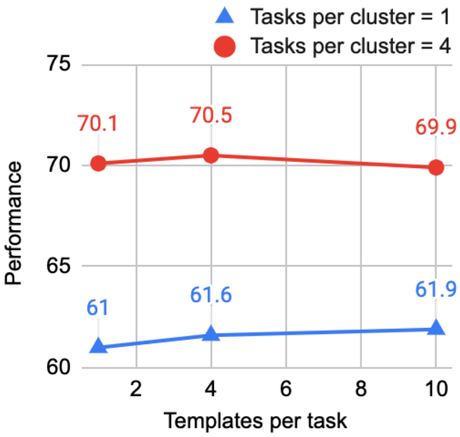


Figure 7: Effect of tasks per cluster and templates per task on performance on three held-out clusters: NLI, commonsense reasoning, and open-domain QA. Adding more tasks per cluster substantially improves performance. Using more templates per task, however, only had a very small effect on performance, which disappeared when there were sufficient tasks per cluster.

B Appendix: Task Details

This appendix provides the details of each dataset that we use in this paper. We group datasets into one of the following clusters:

- **Natural language inference** (7 datasets) concerns how two sentences relate, typically asking, given a first sentence, whether a second sentence is true, false, or possibly true.
- **Reading comprehension** (7 datasets) tests the ability to answer a question when given a passage that contains the answer.
- **Commonsense reasoning** (4 datasets) evaluates the ability to perform physical or scientific reasoning with an element of common sense.
- **Sentiment analysis** (4 datasets) is a classic NLP task aims to understand whether a piece of text is positive or negative.
- **Open-domain QA** (3 datasets) asks models to answer questions about the world without specific access to information that contains the answer.
- **Paraphrase detection** (3 datasets) asks a model to determine whether two sentences are semantically equivalent.[7]
- **Coreference resolution** (3 datasets) tests the ability to identify expressions of the same entity in some given text.
- **Reading comprehension with commonsense** (2 datasets) combines elements of both reading comprehension with commonsense.
- **Data to text** (4 datasets) tests the ability to describe some structured data using natural language.
- **Summarization** (11 datasets) asks models to read a piece of text and generate an abbreviated summary of it.
- **Translation** (6 languages) is the task of translating text from one language into a different language.

For all tasks, our finetuning and evaluation code uses tensorflow datasets (TFDS) to load and process datasets. Regarding the number of training examples per dataset, we limited the training set size per dataset to 30,000 so that no dataset dominated the finetuning distribution. When a test set with labels was available in TFDS, we used it; otherwise, we used the TFDS validation set as our test set, splitting the training set into a train and dev set. On the following pages, we show example inputs and outputs for all evaluation tasks. See our code for the templates for tasks that were used for instruction-tuning but not for evaluation.

[7] Although paraphrasing can be seen as positive entailment in both directions, it has been distinct from NLI in the academic literature.

B.1 Natural Language Inference

INPUT
Joey Heindle (born 14 May 1993 in Munich) is a German singer. He is best known for winning the seventh season of the game show Ich bin ein Star – Holt mich hier raus! and finishing in 5th place in season 9 of Deutschland sucht den Superstar, despite universally negative reviews from the jury each week.

Based on the paragraph above can we conclude that "Joey Heindle was highly disliked by people on television."?

OPTIONS:
- Yes
- It's impossible to say
- No

TARGET
Yes

Table 6: Example input and target for Adversarial NLI (ANLI). ANLI (Nie et al., 2020) is a large-scale NLI benchmark with adversarial examples collected iteratively with a human and model in the loop. The task is to determine whether a hypothesis is entailed by a premise (entailment, not entailment, or impossible to say). There are three rounds, R1–R3. Of the three training sets with 16,946, 45,460, and 100,459 examples, we use 16,946, 30,000, and 30,000 for train and 200 from each of the three TFDS validation sets for dev. We use the TFDS "test" sets of 1,000, 1,000, and 1,200 examples as our test set for reporting numbers.

INPUT
A: so I watch the fish, you know. Whatever I can do to keep myself occupied. I like to have the TV on, because that usually keeps me, um, more occupied. It kind of takes the time away and I don't realize, that's really the only time I ever watch TV, is when I'm on the bike. and then usually after I'm done riding the bike, just to cool myself down, I usually take a walk, you know, and that just kind of uh, gets me, you know, to where I'm not quite as tired I guess. But it's definitely a task. B: You think so? A: I can't say that I really enjoy it.

Based on the paragraph above can we conclude that "she really enjoys it"?

OPTIONS:
- Yes
- No
- It's impossible to say

TARGET
No

Table 7: Example input and target for Commitment Bank (CB). CB (de Marneffe et al., 2019) is a corpus of texts in which a hypothesis is extracted from a premise, and the task is to determine whether the hypothesis is entailed by the premise (entailment, not entailment, or impossible to say). Of the training set with 250 examples, we use 200 for train and 50 for dev. We use the TFDS validation set of 56 examples as our test set for reporting numbers.

INPUT After years of study, the Vatican’s doctrinal congregation has sent church leaders a confidential document concluding that "sex-change" procedures do not change a person’s gender in the eyes of the church. Based on the paragraph above can we conclude that "Sex-change operations become more common."? OPTIONS: - yes - no
TARGET no

Table 8: Example input and target for Recognizing Textual Entailment (RTE). RTE (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009) asks whether a second sentence is entailed by a first (binary, either entailed or not entailed). Of the training set with 2490 examples, we use 2,290 for train and 200 for dev. We use the TFDS validation set of 277 examples as our test set for reporting numbers.

B.2 Reading Comprehension

INPUT

There are four ways an individual can acquire Canadian citizenship: by birth on Canadian soil; by descent (being born to a Canadian parent); by grant (naturalization); and by adoption. Among them, only citizenship by birth is granted automatically with limited exceptions, while citizenship by descent or adoption is acquired automatically if the specified conditions have been met. Citizenship by grant, on the other hand, must be approved by the Minister of Immigration, Refugees and Citizenship.

Can we conclude that can i get canadian citizenship if my grandfather was canadian?

OPTIONS:
- no
- yes

TARGET

no

Table 9: Example input and target for Boolean Questions (BoolQ). BoolQ Clark et al. (2019a) asks a yes/no question based on a passage and a question. Of the training set with 9,427 examples, we use 9,227 for train and 200 for dev. We use the TFDS validation set of 3,270 examples as our test set for reporting numbers.

INPUT

Imagine you are standing in a farm field in central Illinois. The land is so flat you can see for miles and miles. On a clear day, you might see a grain silo 20 miles away. You might think to yourself, it sure is flat around here. If you drive one hundred miles to the south, the landscape changes. In southern Illinois, there are rolling hills. Why do you think this is? What could have caused these features? There are no big rivers that may have eroded and deposited this material. The ground is capable of supporting grass and trees, so wind erosion would not explain it. To answer the question, you need to go back 12,000 years. Around 12,000 years ago, a giant ice sheet covered much of the Midwest United States. Springfield, Illinois, was covered by over a mile of ice. Its hard to imagine a mile thick sheet of ice. The massive ice sheet, called a glacier, caused the features on the land you see today. Where did glaciers go? Where can you see them today? Glaciers are masses of flowing ice.

Question: "How big were the glaciers?"

Response: "One mile"

Does the response correctly answer the question?

OPTIONS:
- no
- yes

TARGET

yes

Table 10: Example input and target for Multi-Sentence Reading Comprehension (MultiRC). MultiRC Khashabi et al. (2018) asks an open-ended question given a paragraph that contains the answer. Of the training set with 27,243 examples, we use 27,043 for train and 200 for dev. We use the TFDS validation set of 4,848 examples as our test set for reporting numbers.

INPUT soil is a renewable resource for growing plants A plant that needs to expand will be able to have an endless resource in OPTIONS: - dirt - pesticides - pay - beans
TARGET dirt

Table 11: Example input and target for Openbook Question Answering (OBQA). OBQA (Mihaylov et al., 2018) asks 4-way multiple choice questions based facts. Of the training set with 4,957 examples, we use all for train and 200 in the TFDS validation set of 500 examples for dev. We use the TFDS test set of 500 examples as our test set for reporting numbers.

B.3 COMMONSENSE REASONING

INPUT
I packed up my belongings. What is the cause?

OPTIONS:
- I was hunting for a new apartment.
- I was moving out of my apartment.

TARGET
I was moving out of my apartment.

Table 12: Example input and target for Choice of Plausible Alternatives (COPA). COPA (Roemmele et al., 2011) is a causal reasoning task that asks to infer either a cause of effect of a premise from two choices. Of the training set with 400 examples, we use 350 for train and 50 for dev. We use the TFDS validation set of 100 examples as our test set for reporting numbers.

INPUT
What happens next in this paragraph?

Once the rope is inside the hook, he begins moving up the wall but shortly after he stops and begins talking. The male then begins talking about the clip again and goes back up the wall. as he
OPTIONS:
- progresses, there are hooks everywhere on the wall and when he gets near them, he puts his rope inside of it for support and safety.
- changes time, an instant replay of his initial move is shown a second time.
- continues to talk, another male speaks about the move and shows another closeup of the plex by the male.
- continues, other people start to arrive and begin to hang out with him as he makes a few parts of the rope.

TARGET
progresses, there are hooks everywhere on the wall and when he gets near them, he puts his rope inside of it for support and safety.

Table 13: Example input and target for Commonsense Sentence Completion (HellaSwag). HellaSwag (Zellers et al., 2019) tests for sentence completion that requires common sense, asking for the most probable ending given four contexts. Of the training set with 39,905 examples, we use 30,000 for train and 200 for dev. We use the TFDS validation set of 10,042 examples as our test set for reporting numbers.

INPUT
Here is a goal: Remove smell from garbage disposal.

How would you accomplish this goal?

OPTIONS:
- Create soda ice cubes and grind through disposal.
- Create vinegar ice cubes and grind through disposal.

TARGET
Create vinegar ice cubes and grind through disposal.

Table 14: Example input and target for Physical Question Answering (PiQA). PiQA (Bisk et al., 2020) is a commonsense QA benchmark for naive physics reasoning, where a solution to a goal must be selected from two choices. Of the training set with 16,113 examples, we use 16,013 for train and 100 for dev. We use the TFDS validation set of 1,838 examples as our test set for reporting numbers.

INPUT Caroline never drinks carbonated beverages. Her friends pick on her because of it. One day they challenged her to drink a soda. Caroline wanted to win the challenge. Predict the next sentence. OPTIONS: - Caroline refused to open the soda. - Caroline opened the soda and drank it all in one gulp!
TARGET Caroline opened the soda and drank it all in one gulp!

Table 15: Example input and target for The Story Cloze Test (StoryCloze). StoryCloze (Mostafazadeh et al., 2016) is a commonsense reasoning framework for story generation, where a system chooses the correct ending to a four-sentence story. We use the 2016 version on TFDS. Of the validation set with 1,871 examples (no training set is available), we use 1,671 for train and 200 for dev. We use the TFDS test set of 1,871 examples as our test set for reporting numbers.

B.4 OPEN-DOMAIN QA

INPUT What season is the Northern Hemisphere experiencing when it is tilted directly toward the Sun? OPTIONS: - fall - winter - spring - summer
TARGET summer

Table 16: Example input and target for The AI2 Reasoning Challenge (ARC). ARC (Clark et al., 2018) asks grade-school level 4-way multiple choice science questions. There is a challenge set and an easy set, where the challenge set questions were answered incorrectly by both a retrieval-based algorithm and a co-occurrence algorithm. Of the training sets with 1,119 examples (challenge) and 2,251 (easy), we use we use 919 and 2,051 respectively for train and 200 each for dev. We use the TFDS test sets of 1,172 and 2,376 examples respectively as our test set for reporting numbers.

INPUT Question: who is the girl in more than you know?? Answer:
TARGET Romi Van Renterghem.

Table 17: Example input and target for Natural Questions (Open) (NQ). NQ (Lee et al., 2019; Kwiatkowski et al., 2019) asks for an open-ended answer given a question, where all questions can be answered using the contents of Wikipedia. Of the training set of 87,925 examples, we use 30,000 for train and 200 for dev. We use the TFDS validation set of 3,610 examples as our test set for reporting numbers.

INPUT Please answer this question: Henry Croft, an orphan street sweeper who collected money for charity, is associated with what organised charitable tradition of working class culture in London, England?
TARGET pearly kings and queens

Table 18: Example input and target for Trivia Question Answering (TriviaQA). TriviaQA Joshi et al. (2017) includes question-answer pairs authored by trivia enthusiasts. Of the training set of 87,622 examples, we use 30,000 for train and 200 for dev. We use the TFDS validation set of 11,313 examples as our test set for reporting numbers.

B.5 Coreference Resolution

INPUT

How does the sentence end?

Elena wanted to move out of her parents fast but Victoria wanted to stay for a while,

OPTIONS:
- Elena went to school.
- Victoria went to school.

TARGET

Victoria went to school.

Table 19: Example input and target for Adversarial Winograd Schema Challenge (Winogrande). Winogrande (Sakaguchi et al., 2020) tests for coreference resolution by asking a model to fill in a masked token in a sentence by choosing an entity from two options. Of the 40.4k examples in the XL training set, we use 30,000 for train and 200 for dev. We use the TFDS validation set of 1,267 as our test set for reporting numbers.

INPUT

Jane knocked on Susan's door, but there was no answer.
OPTIONS:
- Jane was out.
- Susan was out.

TARGET

Susan was out.

Table 20: Example input and target for Winograd Schema Challenge (WSC273). WSC273 (Levesque et al., 2012) tests for coreference resolution by asking a model to complete the sentence in a fashion that requires understanding the entities in the sentence. Of the 0 examples in the training set (WSC273 is test-set only), we use none for train and none for dev. We use the TFDS test set as our test set for reporting numbers.

B.6 Reading Comprehension with Commonsense

INPUT
Complete the passage.

(CNN) – At first glance, "The Flat" might seem like an episode of "Hoarders," Israeli-style. The documentary film opens after an elderly woman dies in Tel Aviv. Her grandchildren assemble to clean out her apartment, packed with dusty books, vintage clothing (dozens of pairs of fancy gloves, for instance), enough purses to stock a department store, jewelry, mementoes and closets full of knickknacks. But buried among the detritus they chance upon something remarkable – mysterious papers linking the grandparents to an important Nazi figure. How could such ardent Zionists, who left their native Germany in the early 1930s, have been involved with an SS official like Leopold von Mildenstein?

What I found out was this journey, the Nazi (

OPTIONS:
- Arnon Goldfinger) and his wife were accompanied by my grandparents," Goldfinger told CNN.
- CNN) and his wife were accompanied by my grandparents," Goldfinger told CNN.
- Germany) and his wife were accompanied by my grandparents," Goldfinger told CNN.
- Israeli) and his wife were accompanied by my grandparents," Goldfinger told CNN.
- Leopold von Mildenstein) and his wife were accompanied by my grandparents," Goldfinger told CNN.
- Nazi) and his wife were accompanied by my grandparents," Goldfinger told CNN.
- SS) and his wife were accompanied by my grandparents," Goldfinger told CNN.
- Tel Aviv) and his wife were accompanied by my grandparents," Goldfinger told CNN.
- The Flat) and his wife were accompanied by my grandparents," Goldfinger told CNN.
- Zionists) and his wife were accompanied by my grandparents," Goldfinger told CNN.

TARGET
Leopold von Mildenstein) and his wife were accompanied by my grandparents," Goldfinger told CNN.

Table 21: Example input and target for Reading Comprehension with Commonsense Reasoning (ReCoRD). ReCoRD (Zhang et al., 2018) asks for the answer to a cloze-style question where an entity is masked out. Of the the training set of 100,730 examples, we use 30,000 for train and 200 for dev. We use the TFDS validation set of 10,000 examples as our test set for reporting numbers.

B.7 Translation (7 languages)

INPUT
Here the largest town of the district is located: Nordenham , lying opposite to Bremerhaven at the Weser mouth.

Translate to German

TARGET
An der B 211 befindet sich in Loyermoor der so genannte “Geest-Abbruch”, der eine Höhendifferenz von gut 30 Meter überbrückt.

Table 22: Example input and output for translation. This example is from WMT’16 English–German; all languages use the same translation templates.