A Systematic Investigation of Commonsense Understanding in Large Language Models

Xiang Lorraine Li[†] * **Adhi Kuncoro**[‡] **Cyprien de Masson d'Autume**[‡]
Phil Blunsom[‡] **Aida Nematzadeh**[‡]
[†] University of Massachusetts Amherst
[‡] DeepMind
xiangl@cs.umass.edu, {akuncoro,cyprien,pblunsom,nematzadeh}@google.com

Abstract

Large language models (*e.g.*, Brown et al., 2020; Patwary et al., 2021) have shown impressive performance on many natural language processing (NLP) tasks in a zero-shot setting. We ask whether these models exhibit commonsense understanding – a critical component of NLP applications – by evaluating models against four commonsense benchmarks. We find that the impressive zero-shot performance of large language models is mostly due to existence of dataset bias in our benchmarks. We also show that the zero-shot performance is sensitive to the choice of hyper-parameters and similarity of the benchmark to the pre-training datasets. Moreover, we did not observe substantial improvements when evaluating models in a few-shot setting. Finally, in contrast to previous work, we find that leveraging explicit commonsense knowledge does not yield substantial improvement.

1 Introduction

Large language models (with more than 1 billion parameters) perform well on a range of natural language processing (NLP) tasks in zero- and few-shot settings, without requiring task-specific supervision (*e.g.*, Radford et al., 2019; Brown et al., 2020; Patwary et al., 2021). This observation suggests that given enough capacity, language models may extract the knowledge required to perform well on these NLP tasks from raw text, simply using the transformer architecture (Vaswani et al., 2017) and auto-regressive language modeling objective. Consequently, various recent efforts have focused on pre-training evergrowing language models exceeding even 350 billion parameters (Brown et al., 2020; Patwary et al., 2021).

Tasks that require commonsense understanding are of special interest in such zero- and few-shot evaluation paradigms because it is hard to collect supervised commonsense datasets given the vast, diverse, and growing nature of commonsense knowledge (Elazar et al., 2021). Despite the importance of commonsense understanding in NLP systems (McCarthy et al., 1960; Gunning, 2018; Liu and Singh, 2004), whether commonsense knowledge can be acquired by large language models and through the language modeling objective remains an open question.

To answer this question, we evaluate pre-trained language models against multiple-choice question-answering benchmarks, examining different types of commonsense knowledge, such as social, physical, and temporal (Bisk et al., 2020; Sakaguchi et al., 2020; Zellers et al., 2019b; Sap et al., 2019b). We focus on zero- or few-shot evaluation that enables us to examine the commonsense understanding capacity of pre-trained language models. We compare zero- and few-shot performance of the models with that of different baselines; in particular, we consider an *Answer-only* baseline where a model choose an answer based on the likelihood of the answer choices alone *without* considering the question. The high performance of the *Answer-only* baseline is a sign of *dataset bias* in a benchmark – questions are not required to solve an example.

First, we evaluate our largest model (with 7 billion parameters) against commonsense benchmarks in a zero-shot way: we find that zero-shot performance is still far from the state-of-the-art fine-tuned results. In fact, zero-shot performance is always closer to that of the Answer-only baseline. We also observe a correlation between zero-shot performance and the similarity of a benchmark to the pre-training data (measured as the perplexity of the benchmark under a pre-trained model). Our results show that the zero-shot performance of pre-trained language models on commonsense benchmarks is mostly attributed to the dataset bias in the benchmarks; it also highlights the need of reporting

* Work done during internship with DeepMind.

strong baselines in our evaluations which is missing from some of the recent work (Radford et al., 2019; Brown et al., 2020; Patwary et al., 2021).

Next we investigate to what extent hyper-parameters (such as the prompt format) and increasing model size impact the zero-shot performance. In all commonsense benchmarks, we see a gap between the zero-shot performance of the worst and best hyper-parameter settings (ranging between 2 to 19 accuracy points). This result shows that for commonsense understanding tasks, it is best not to take pre-trained language models as off-the-shelf tools. When increasing model size, across benchmarks, we observe improvements on *both* zero-shot performance and the Answer-only baseline, suggesting that larger models are better in exploiting the surface cues.

Finally, we examine if pre-trained language models benefit from adding more examples in a few-shot setting or leveraging knowledge extracted from existing commonsense knowledge bases. We do not observe a substantial gain from the few-shot evaluation compared to the zero-shot one. Adding commonsense knowledge also does not yield notable improvements, showing that our language models cannot leverage the relevant knowledge when it is simply added to the prompt.

It is an exciting time for language research in both academia and industry, with various efforts working on pre-training stronger language models. However, to better understand the goodness of these models, we need to compare them with strong baselines. Moreover, we need to compare different models under similar hyper-parameter settings given that the choice of parameters can substantially impact model performance even in a zero-shot evaluation setting. We hope our work encourages the community to consider stronger evaluation protocols for pre-trained language models.

2 Experimental Setting

We evaluate the performance of pre-trained language models against commonsense benchmarks in zero-shot and few-shot settings (without updating models' parameters). In this section, we discuss the benchmarks, models, baselines and other experimental settings.

2.1 Commonsense Benchmarks

Commonsense knowledge can be divided into different categories such as physical (*e.g.*, a car is heavier than an apple), social (*e.g.*, a person will feel happy after receiving gifts), and temporal (*e.g.*, cooking an egg takes less time than baking a cake). Various benchmarks have been proposed to test these different types of commonsense knowledge (*e.g.*, Zellers et al., 2019b; Sakaguchi et al., 2020; Sap et al., 2019b; Bisk et al., 2020; Lin et al., 2020; Boratko et al., 2020). To examine if pre-trained language models exhibit commonsense knowledge, we are interested in benchmarks that are representative of different types of commonsense.

Commonsense benchmarks broadly consist of two types of tasks: (a) the multiple-choice selection evaluation (Zellers et al., 2019a,b; Sap et al., 2019b; Bisk et al., 2020) where a model needs to choose the right answer from a list of candidates including the correct choice and a few incorrect ones; (b) the generative evaluation (Boratko et al., 2020; Lin et al., 2020, 2021) which requires a model to generate an answer given a question and some additional context. We focus on multiple-choice selection benchmarks since they provide a more reliable automatic metric (*i.e.*, accuracy) compared to those used to evaluate language generation (*e.g.*, BLEU). A thorough evaluation of natural language generation requires considering human annotations in addition to automatic metrics such as BLEU (Papineni et al., 2002; Lee et al., 2021). However, human evaluation is also imperfect (Clark et al., 2021) adding to the difficulty of evaluating common sense in a generation setup.

We use four benchmarks (listed in Table 1) to examine commonsense understanding in language models which we briefly describe below. We use the validation split of the selected benchmarks.

HellaSwag (Zellers et al., 2019b) mainly evaluates physical, grounded, and temporal commonsense knowledge. Given a short (four-sentence) story, the task is to choose the correct ending from four candidate sentences. The stories are either video captions taken from AcitivityNet (Heilbron et al., 2015) or WikiHow passages (Koupaee and Wang, 2018). When evaluated under a pre-trained language model, negative candidates can be easy to distinguish from positive ones if they are unlikely to occur with the question context. Zellers et al. (2019b) use an adversarial filtering step to remove such easy negative candidates.

WinoGrande (Sakaguchi et al., 2020) is a co-reference resolution benchmark that mainly examines physical and social commonsense knowledge.

	Choices	Reasoning Types	Questions
HellaSwag (Zellers et al., 2019b)	4	Temporal, Physical, etc	10042
WinoGrande (Sakaguchi et al., 2020)	2	Social, Physical, etc	1267
Social IQa (Sap et al., 2019b)	3	Social	1954
PIQA (Bisk et al., 2020)	2	Physical	1838

Table 1: Benchmark Statistics. For each benchmark, "Choices" and "Questions" show the number of candidate answers for each question and the number of questions in the validation split, respectively.

Each data point consists of a sentence (*e.g.*, "The trophy did not fit the suitcase because it is too big.") and two candidate *entities* (*e.g.*, "trophy" or "suitcase"). The task is to choose the correct entity for the pronoun in a given sentence, such as "it" in the this sentence.

Social IQa (Sap et al., 2019b) focuses on evaluating social commonsense, and in particular on theory of mind – the capacity to reason about others' mental states, such as beliefs (Flavell, 2004). Given context sentences and a corresponding question, the task is to choose the correct response from three answer candidates. Human annotators use the ATOMIC knowledge bases (Sap et al., 2019a) to create context sentence and questions; the answers are provided by additional human annotators.

PIQA (Bisk et al., 2020), short for physical interaction question answering, mainly covers the physical aspects of commonsense. Each example in this benchmark consists of a task and two alternative solutions to finish the task, one of which is correct. The tasks are curated from a website[1] with instructions for everyday tasks (*e.g.*, separating egg yolks from eggs) and the solutions are provided by human annotators.

2.2 Pre-trained Language Models

The pre-trained models are autoregressive language models with an architecture similar to GPT3 (Brown et al., 2020). Similarly to T5 model (Raffel et al., 2019), we train our models using the cleaned version of Common Crawl corpus (C4), around 800 GB of data. Our largest model, with 32 transformer layers and 7 billion parameters, has a similar number of parameters to the open-sourced GPT-J model (Wang and Komatsuzaki, 2021), and is much larger than the largest GPT2-XL model with only 1.5 billion parameters (Radford et al., 2019).

We evaluate a pre-trained language model against multiple-choice selection benchmarks where given a question, the model needs to select one of the candidate answer choices. To do so, we calculate a score for each answer choice under the model, and select the answer with the highest score:

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in Y(\mathbf{x})} s_\theta(\mathbf{y}|\mathbf{x})$$

where $\mathbf{x}$ is the question or prompt, $Y(\mathbf{x})$ denotes the possible answer choices given the question, and s_θ gives the score of an answer choice $\mathbf{y}$ given $\mathbf{x}$ under the pre-trained model with parameters θ. Examples of $\mathbf{x}$ and $\mathbf{y}$ for each dataset are shown in Table 2.[2] For Social IQa, we convert questions to natural text using rules given in Shwartz et al. (2020) since the natural text yields better performance as discussed in §4.

To calculate the score of an answer choice ($s()$), unless otherwise stated, we use *entropy (token-level log probability)* – the answer sequence log probability under the language model divided by its length:

$$s(\mathbf{y}|\mathbf{x}) = \frac{\sum_{i=0}^{\|\mathbf{y}\|} log(p(y_i|x, y_0 ... y_{i-1}))}{\|\mathbf{y}\|} \quad (1)$$

This formulation reduces the impact of sequence length *i.e.*, longer answers might have lower probability (Stahlberg and Byrne, 2019). The recent GPT3 language model (Brown et al., 2020) also uses this score for zero-shot evaluation.

2.3 Baselines

We compare the performance of pre-trained language models with different baselines. A simple baseline is to randomly select an answer from the answer candidate choices, where the chance of selecting the correct answer is $\frac{1}{number\ of\ choices}$ for each

[1] https://www.instructables.com/

[2] For Social IQa, we concatenate context sentence and question together to form the prompt x.

Dataset	Prompt: x	Answer: y
HellaSwag	A woman is outside with a bucket and a dog. The dog is running around trying to avoid a bath. She	gets the dog wet, then it runs away again.
WinoGrande	None	The home that my parents had when I was in school was a lot nicer than my house now because the **house** is trashy.
Social IQa	Jordan was in charge of taking the food on the camping trip and left all the food at home. Jordan felt	horrible that he let his friends down on the camping trip.
PIQA	Make Halloween lanterns.	Draw ghost faces on empty milk bottles, put a candle in each one.

Table 2: An example prompt x and its scored answer y in different benchmarks. In Social IQa, the input questions are converted into sentences.

question. We refer to this baseline as the *Random Baseline*. We also experimented with two other baselines by either choosing the majority label from the training data or the longest answer as the prediction. We do not report those results since they perform similarly to the Random Baseline.

We also consider an *Answer-only Baseline* where we evaluate the answer options under the pre-trained models *without* considering the question. This baseline show to what extent a pre-trained model uses the question to make a prediction as opposed to relying on the likelihood of answer choices in the pretraining data. We calculate the entropy of the answer-choices $\mathbf{y}$ in Table 2 for HellaSwag, WinoGrande and PIQA. For WinoGrande, we calculate the entropy of the answer choices. Without any dataset bias, we anticipate answer probabilities under a model to be independent of answer correctness, and a performance similar to the Random Baseline. Similar hypothesis-only baselines are well studied for natural language inference problems to reveal the dataset bias (McCoy et al., 2019). Trichelair et al. (2019) study the Answer-only Baseline on one commonsense benchmark, SWAG (Zellers et al., 2019a).

3 Commonsense in Language Models

Previous work suggests that large language models, to some extent, exhibit commonsense knowledge (Trinh and Le, 2018; Brown et al., 2020; Shwartz et al., 2020). We examine the extent to which large language models capture commonsense by evaluating them in a zero-shot way against the benchmarks discussed in §2.1. Fig. 1 gives the accuracy of our 7B-parameter model, Random and Answer-only Baselines, and the state-of-the-art (SOTA) results from fine-tuned models.

Zero-shot performance. At the first glance, the zero-shot performance is better than the Random Baseline for all benchmarks (compare Rand and ZS in Fig. 1); however, the gap between the stronger Answer-only Baseline and zero-shot performance is much smaller (compare Answer and ZS). While this performance gap (between zero-shot and Answer-only Baseline) is notable for HellaSwag and Social IQa (greater than 10 point accuracy), it is small for WinoGrande and PIQA. We also observe that the gap between fine-tuned SOTA and zero-shot performance is still large for all benchmarks, with WinoGrande and Social IQa having the largest gaps, suggesting that these two benchmarks are more challenging for pre-trained models. Finally, the zero-shot performance is *always* closer to the Answer-only Baseline than to the fine-tuned SOTA.

Dataset bias. As shown in Fig. 2, the performance gap between Random and Answer-only Baselines is large for HellaSwag and PIQA, with relative accuracy improvements of 26% and 21%, respectively. This large performance gap highlights the existing dataset bias in these benchmarks: the correct answer can be selected by a model without considering its question and thus without commonsense reasoning that is grounded on the input questions. On the other hand, the gap between Random and Answer-only Baselines is small for WinoGrande and Social IQa; as a result, the performance on these benchmarks better captures the goodness of pre-trained models in commonsense understanding. Interestingly, these two benchmarks are the hardest for the model – have the largest difference

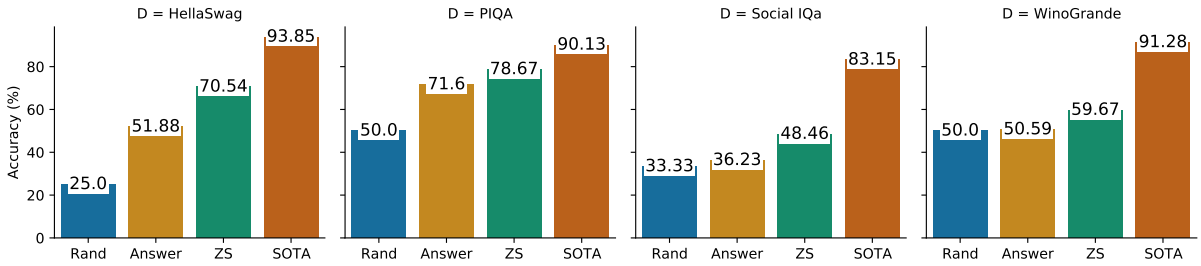


Figure 1: Random Baseline (Rand), Answer-only Baseline (Answer), zero-shot (ZS), and fine-tuned state-of-the-art (SOTA) accuracy for each benchmark.

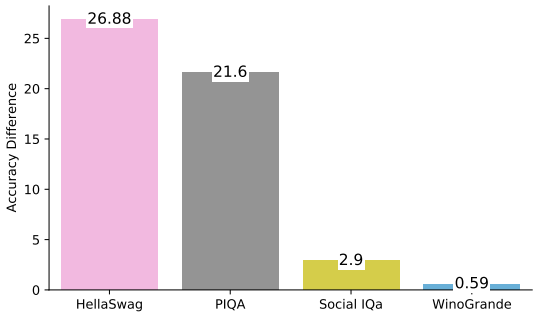


Figure 2: The performance gap between Answer-only and Random Baselines for each benchmark.

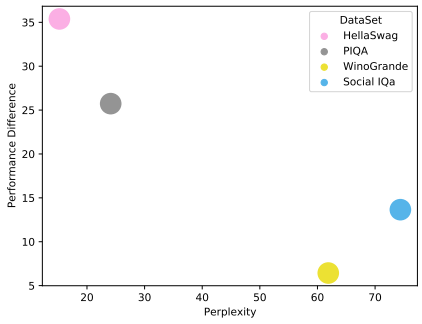


Figure 3: The relationship between model performance and perplexity. The x axis is the average perplexity of examples in a benchmark under the 1B model; the y axis is the performance difference between zero-shot and the Random Baseline.

between zero-shot and fine-tuned performance.

Similarity to pretraining data. We investigate to what extent model performance is due to the similarity between a given benchmark and the pre-training data. To do so, for a given benchmark, we calculate the perplexity of each question and its correct answer choice under the pre-trained language model[3]. To measure model performance, we calculate the gap between the zero-shot and Random Baseline performance, because benchmarks have different number of answer choices and thus different Random Baselines (see Fig. 1). As shown in Fig. 3, we see that lower perplexity often correlates with higher zero-shot performance. As a result, the performance is not only a function of the model's commonsense understanding, but also a function of dataset similarity between the benchmark and the pretraining data.

In summary, we find that although the large language model performs better than a strong baseline, its improvement is not significant for all benchmarks, and its zero-shot performance is still far from the SOTA results. Moreover, we show that given the existing (and sometimes inevitable) biases in some benchmarks, it is important to compare the models with strong baselines which are sometimes omitted from recent work (*e.g.*, Zhou et al., 2020; Brown et al., 2020). Finally, the good performance of a commonsense benchmark can be a result of its similarity to the pre-training dataset as opposed to a model's commonsense reasoning ability: we observe that the performance of commonsense benchmarks correlates with their perplexity under the pre-trained language model.

[3]We use the 1B model mentioned in §5.

4 Robustness of Zero-Shot Performance

Different choices of hyper parameters (such as the prompt's format or the score function) can impact the zero-shot performance of language models and result in different conclusions about their commonsense understanding ability. Moreover, the lack of a standardized zero-shot setting for evaluating language models makes direct comparisons between papers difficult. We perform a set of experiments to shed light on the effect of these choices (listed in Fig. 4) on the performance of our models on commonsense benchmarks.

Score Function:
- Token-level log probability
- Sentence-level log probability
- Mutual Information

Prompt Format:
- "question" + "answer"
- Add special token [Question], [Ans] for "question" and "answer"
- Converted "question" to natural sentence according to rules.

Scored Text:
- Question, answer concatenation.
- Answers conditioned the question.

Figure 4: Hyper-parameters that affect the zero-shot performance.

Score functions. Previous work uses different score functions to assess the plausibility of each answer choice given a question (Brown et al., 2020; Shwartz et al., 2020; Bosselut et al., 2021; Holtzman et al., 2021); hence, the results from previous work are not directly comparable. We investigate the extent to which the choice of score functions impacts models' performance. In addition to entropy (token-level log probability, defined in §2.2), we also experimented with two other score functions: *sequence log probability* is the log probability of the answer choice $\mathbf{y}$ given the question $\mathbf{x}$, where i is the i-th token in the answer sequence.

$$s(\mathbf{y}|\mathbf{x}) = \sum_{i=0}^{\|\mathbf{y}\|} log(p(y_i|\mathbf{x}, y_0...y_{i-1})) \quad (2)$$

We also consider the *point-wise mutual information* between the probability of the answer choices alone and the probability of the answer choices given the questions which is another widely-used score function (Bosselut et al., 2021; Holtzman et al., 2021). This metric assesses whether the question adds additional information to the answers, as commonsense reasoning should be established within the context of the question. Since this score function removes the probability of answer options in its calculation, it can perform worse than scores (such as entropy) that use answer probabilities and thus capture the bias towards answer choices alone (see the Answer-only Baseline in §3).

$$s(\mathbf{y}|\mathbf{x}) = PMI(\mathbf{y}, \mathbf{x}) = log\frac{p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})} \quad (3)$$

Prompt format. Another important factor to task performance is a language model's input formatting (*i.e.*, the prompt format). We consider a few choices that are listed in Fig. 4. In particular, in addition to the concatenation of the question and the answer, we experimented with adding special symbols to specify the question and the answer. Moreover, for Social IQa, we used a set of predefined rules (taken from Shwartz et al., 2020) to convert the questions to sentences (that are closer to the language model's training data). Finally, we find that, for a given question and answer, having correct lower/upper case and punctuation is important in zero-shot performance; thus, we manually checked all benchmarks to correct for case and punctuation.

	Worst	Best	Gap
HellaSwag	50.8	**70.5**	19.7
PIQA	62.5	**78.7**	16.2
Social IQa	43.9	**48.5**	4.6
WinoGrande	59.7	**62.0**	2.3

Table 3: The performance difference between the worst and best hyper-parameters for each benchmark.

Scored text. The next option is whether the question–answer pair or only the answer option is scored conditioned on the given question, *i.e.*, $s(\mathbf{y}|\mathbf{x})$ or $s(\mathbf{x};\mathbf{y})$, where ; implies text concatenation.

4.1 Does Tuning Hyper-Parameters Matter?

Table 3 shows the performance difference of using the worst versus the best hyper-parameter settings for each benchmark. The details for each setting are described in Appendix A. To sweep over hyper-parameter choices in Fig. 4, instead of considering all combinations of parameters, we iterate the options in one category (*e.g.*, score function) while fixing the parameters in the two other categories.[4]

Overall, we observe a difference between best and worst settings of all commonsense benchmarks; this gap is especially large for HellaSwag and PIQA. This result shows that *large language models do not simply work out of the box for commonsense benchmarks, because the choice of hyper-parameters plays an important role in their performance*. We also find that the choice of the score function plays the most important role in task performance – entropy achieves the best per-

[4]This decision helps us save compute resources by running fewer sweeps, and does not change our conclusions given that we are interested in finding if a variance exists across different settings and not the highest achievable performance.

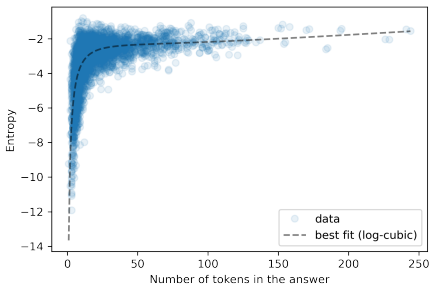


Figure 5: Answer length vs entropy (average log probability across tokens) for PIQA .

formance across most benchmarks, sentence log probability achieved slightly better performance for WinoGrande compared to entropy.; however, as discussed in §3, when using this score, it is important to compare the performance with an Answer-only Baseline. Moreover, converting questions to sentences makes the biggest performance difference for Social IQa compared to other benchmarks. Finally, we find that scoring the answer conditioned on the question works best except for WinoGrande that does not have any questions.

Given our findings, we think that moving forward, the field should establish a unified protocol for performing zero-shot evaluations to enable a fair comparison across papers. Alternatively, when comparing models, it is important to perform similar sweeps for hyper-parameters and report the best setting for each model.

4.2 The Answer-Length Bias

Both sequence log probability and entropy are impacted by the length of the answers for some of the datasets. Fig. 5 shows the relationship between answers' length and their predicted entropy (token-level log probability) for PIQA. Other datasets are shown in Appendix B. We can see that entropy tends to assign higher score to longer answer choices. This pattern holds for 3 out of the 4 datasets, to varying extent. This pattern is often reversed in sequence metrics such as sequence log probability (Koehn and Knowles, 2017). The answer-length bias does not affect the pattern of results we observe in commonsense benchmarks; we also observed that the longest-answer baseline performs similarly to the Random Baseline. However, when using score functions dependent on length, it is important for future work to check for correlations between answer lengths and their correctness.

Model Size	Layers	d_{model}	Heads
44M	8	512	16
117M	12	768	12
400M	12	1536	12
1.3B	24	2048	16
7B	32	4096	32

Table 4: Parameter specifications for different models under the study.

5 Does Increasing Model Size Help?

Recent work has achieved remarkable language modeling progress—in terms of perplexity and downstream task performance—by using *ever-larger* language models (Kaplan et al., 2020; Brown et al., 2020; Patwary et al., 2021). Can increasing language model size also help the model perform better at various commonsense understanding benchmarks? And can we expect to reach the current commonsense SOTA performance—which are currently achieved by fine-tuned models that leverage *task-specific* commonsense annotations—by training ever-larger language models in an unsupervised fashion, and from textual input alone?

Setup. We vary the language model sizes from 44M up to 7B parameters, where the model with 7B parameters—which forms the basis for all our prior experiments—serves as the largest one. We do not study models with more parameters than the 7B models; training multiple, increasingly larger language models—each with substantially more parameters than the 7B model, which is itself already > 4 times larger than the GPT-2 model (Radford et al., 2019)—has prohibitive costs. We expect our findings to generalize to larger models beyond 7B parameters, although we leave it to future work to firmly establish whether this is the case.

Discussion. We present the model size findings in Table 5. On all four benchmarks, the language model's zero-shot performance consistently gets better as we use increasingly larger models. At first glance, this finding is consistent with that of Brown et al. (2020), who show that larger models have better performance at HellaSwag, WinoGrande, and PIQA. But, crucially, we argue that this finding does *not* necessarily mean that larger models are better at commonsense understanding: For HellaSwag and PIQA, the performance of the Answer-only Baseline also increases substantially with model size (Table 5, **Answer-only** column).

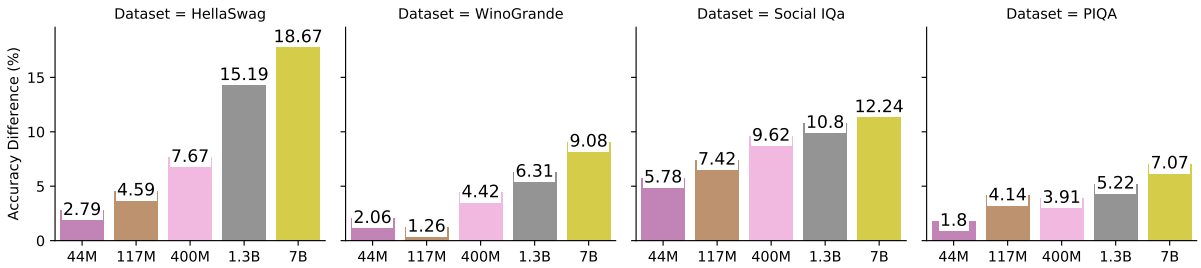


Figure 6: The difference between zero-shot performance and Answer-only baseline.

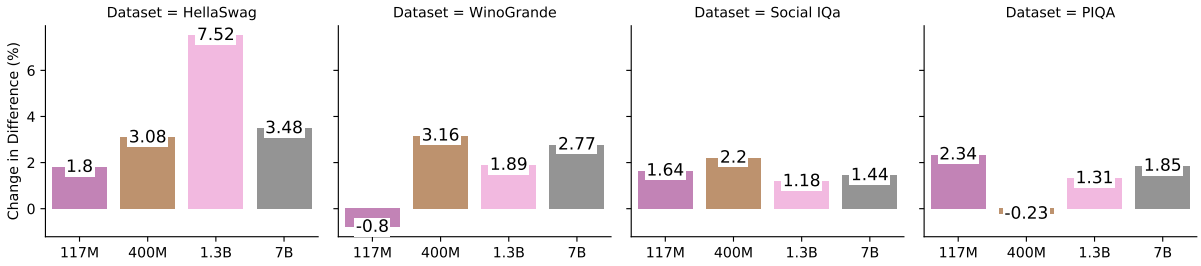


Figure 7: The change in the gap between the zero-shot performance and Answer-only baseline when increasing model size. Each bar in this plot is derived by subtracting two bars in Fig. 6; for example, the 7B data point in this plot is the result of subtracting that of the 1.3B model from the 7B model in Fig. 6.

		Ans	ZS	FS(1)	FS(5)	FS(10)
HellaSwag	**44M**	26.9	**29.6**	29.1	29.3	29.4
	117M	31.0	35.6	35.2	**35.7**	35.2
	400M	36.7	44.4	43.6	**44.9**	44.1
	1.3B	45.2	**60.4**	59.1	59.5	58.8
	7B	51.9	**70.6**	69.2	70.4	69.1
WinoGrande	**44M**	50.4	52.4	**53.0**	52.3	51.8
	117M	49.9	51.1	**53.7**	52.6	52.0
	400M	50.0	**54.5**	53.6	54.3	52.3
	1.3B	50.1	56.4	**56.8**	54.5	54.6
	7B	50.6	59.7	59.7	**61.1**	60.4
Social IQa	**44M**	36.5	**42.3**	41.0	40.9	40.8
	117M	36.2	**43.6**	43.5	42.9	42.6
	400M	36.7	**46.3**	45.2	45.0	45.1
	1.3B	36.2	47.0	46.5	46.9	**48.0**
	7B	36.2	48.5	48.6	49.6	**52.7**
PIQA	**44M**	62.7	**64.5**	64.2	63.6	63.7
	117M	63.7	**67.8**	67.4	66.6	67.3
	400M	67.0	**71.0**	69.9	70.2	70.9
	1.3B	70.5	75.7	75.1	75.7	**75.8**
	7B	71.6	78.7	78.1	**79.2**	79.1

Table 5: Performance of all models across benchmarks under different experimental settings. Ans: Answer-only Baseline; ZS: zero-shot performance; FS(n): few-shot performance where n is the number of examples used.

Hence, for some benchmarks, larger models *also* perform better at exploiting surface cues that enable them to select the correct answer—*without* conducting the appropriate commonsense reasoning based on the provided context and question.

Based on this observation, we reemphasize that, to properly assess commonsense reasoning ability, we should look beyond the zero-shot performance, and instead focus on the *performance difference* between the zero-shot setup and the Answer-only Baseline (§3). We thus plot this performance difference with respect to different model sizes in Fig. 6, based on which we remark on two key observations. First, we observe that larger language models generally have better performance across benchmarks – when increasing model size, the zero-shot performance gains are *more* than the performance gains of the Answer-only Baseline. Nevertheless, the *magnitude* of the improvement from using larger models varies considerably: we observe substantial improvements on HellaSwag, and smaller improvements on WinoGrande, Social IQa and PIQA

Second, in Fig. 7, we report the gains stemming from increasing model size: we plot the change in the zero-shot and Answer-only difference between a model and its immediately preceding smaller model. Crucially, even when model size helps improve the gap between the zero-shot model and the Answer-only Baseline in Fig. 6, we generally observe small gains from increasing model size, which indicates that *simply adding more parameters and training ever-larger models may not help us reach substantially better performance,* although a more comprehensive investigation of even larger models remains within the realm of future work.

	Prompt: x	Answer: y
Answer-only	None	horrible that he let his friends down on the camping trip.
Zero-shot	Jordan was in charge of taking the food on the camping trip and left all the food at home. Jordan felt	horrible that he let his friends down on the camping trip.
Few-shot	Carson was excited to wake up to attend school. Carson did this because they wanted just say hello to friends. \n Jordan was in charge of taking the food on the camping trip and left all the food at home. Jordan felt	horrible that he let his friends down on the camping trip.

Table 6: An example of the prompt and its answer (taken from Social IQa) in different experimental settings: answer-only baseline, zero-shot, and few-shot. The scored answer is the same across conditions.

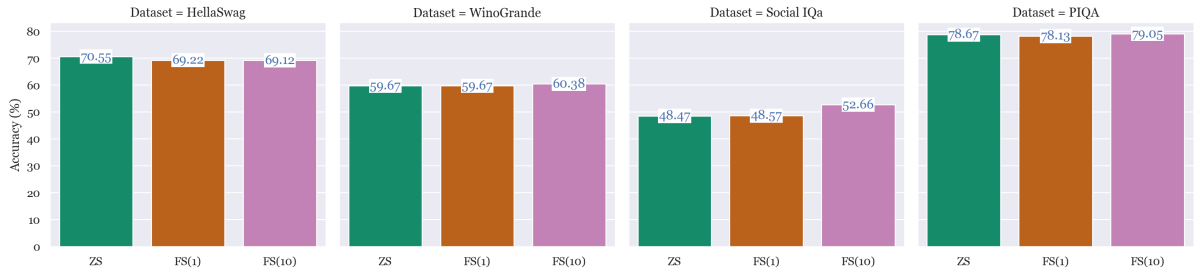


Figure 8: The performance on different benchmarks under three different evaluation settings: zero-shot, few-shot with 1 example, and few-shot with 10 examples.

6 Few-shot Evaluation

Recent work has shown that large language models are able to perform surprisingly well at various downstream tasks in a few-shot fashion (Brown et al., 2020; Patwary et al., 2021). Under this setup, the model is provided with n examples of the downstream task, which are then appended to the language model prefix. Concretely, for the four common-sense benchmarks, we append n examples that include: (i) the question and (ii) the correct answer; these examples—which we obtain from the training split of each benchmark through a random sampling procedure—appear before the evaluated question, as shown in Table 6. This few-shot formulation is indeed an appealing one: It only requires a small number of task-specific examples to get the language model accustomed to what is expected of the task—all *without* the need to update the entire set of model parameters through fine-tuning, which requires large amounts of common-sense labels and incurs significant computation costs. Hence, we ask: To what extent can we improve performance on common-sense benchmarks, by shifting from the zero-shot to the few-shot evaluation protocol?

Discussion. In Fig. 8, we compare the performance of the same 7B model, albeit with three different evaluation protocols: (i) zero-shot, (ii) few-shot with 1 example, and (iii) few-shot with 10 examples, based on which we remark on two key observations. First, on most datasets (HellaSwag, WinoGrande, and PIQA), we do not observe substantial improvement from few-shot evaluation compared to the zero-shot baseline, which does not benefit from any task-specific examples; in fact, model performance with few-shot (1) is sometimes *worse* than the zero-shot model, although we find that the few-shot (10) model mostly outperforms its zero-shot counterpart, albeit by small margins. Second, while few-shot evaluation does not help much for most datasets, the only exception is Social IQa, where we observe that the few-shot (10) model substantially outperforms the zero-shot model by a $> 4\%$ margin. We attribute this to the less natural text of Social IQa; hence adding a few task-specific examples provides informative examples of what is expected of the task.

We remark that our pattern of results is different than that of Brown et al. (2020): they observe larger improvements under the few-shot setup for Wino-

Grande (*i.e.*, 7.5% accuracy gain). But, they similarly find small accuracy improvements for PIQA and HellaSwag (less than %1.5). We attribute this difference to two factors: (i) GPT-3 use a much larger language model with 175B parameters. We thus conjecture that the better performance of language models' few-shot learning ability may be confined to much larger models than the 7B model that we use in this work. (ii) GPT-3 uses a larger number of few-shot samples (n=50), while we limit our experiments to n=10. Except for Social IQa, we observe little improvement from increasing the number of prefix from n=1 to n=10; hence we do not experiment with appending more examples beyond 10.

7 Commonsense Knowledge Base

Given the implicit nature of common-sense knowledge, a language model's pretraining corpora might not contain all of the supporting evidence that is required to solve common-sense understanding questions—a phenomenon widely known as the reporting bias problem (Gordon and Van Durme, 2013). To this end, prior work has proposed the use of external knowledge bases for improving the zero-shot performance of language models on common-sense benchmarks (Bosselut et al., 2021; Bauer and Bansal, 2021). This approach is particularly interesting since the language model is pretrained in the usual fashion, and is only combined with a knowledge base during evaluation. This formulation thus renders an approach compatible with *any* pretrained generative language model, which can then be combined with different kinds of commonsense knowledge bases that only need to be specified at test time. While prior work has shown the effectiveness of this approach over the zero-shot baseline that lacks access to commonsense knowledge bases, we find that the performance of the baseline zero-shot model is, in fact, highly sensitive to the choice of evaluation hyper-parameters (§4). A natural question, therefore, is the following: If we optimize the evaluation hyper-parameters of the baseline zero-shot model carefully, would we still observe the same improvement when we augment the model with common-sense knowledge bases?

Setup. To this end, we replicate prior work by adding common-sense knowledge base entries at test time. To ensure the generality of our findings, we apply this approach to multiple model sizes that we explored in §5. For the knowledge base entries, we consider the pre-extracted knowledge base triples that are made publicly available by Shwartz et al. (2020). We use the same score function as Shwartz et al. (2020), where we choose the highest score for each answer choice among all extracted knowledge base triples, under the same assumption that not all extracted triples are useful for the task:[6]

$$s_{kg}(\mathbf{y}) = \sum_{\mathbf{t} \in T} s(\mathbf{x}; \mathbf{y}; \mathbf{t}) \approx max_{\mathbf{t} \in T}\, s(\mathbf{x}; \mathbf{y}; \mathbf{t}),$$

where $s(\mathbf{x}; \mathbf{y}; \mathbf{t})$ denotes the entropy (token-level log probability) of the concatenation of question $\mathbf{x}$, answer choice $\mathbf{y}$ and the extracted knowledge base triple $\mathbf{t}$. T denotes the set of all extracted common-sense knowledge triples.

We summarize our findings on Social IQa in Table 7 which has the highest gap between the zero-shot and SOTA performance, and leave the extension to other common-sense benchmarks to future work. We compare our results with those of Shwartz et al. (2020), who used GPT-2 as the base model. Our results in Table 7 provide an interesting contrast to the findings of Shwartz et al. (2020): Our baseline zero-shot model with 1.3B parameters achieves an accuracy of 46.98% on Social IQa, substantially outperforming the reported GPT-2 result of Shwartz et al. (2020)—which achieves 41.1%—despite the fact that GPT-2 has more parameters (1.5B vs our 1.3B model). In fact, the per-

	ZS	w/t Comet	w/t Atomic	w/t CN
44M	42.3	**42.9**	42.3	40.6
117M	43.6	**44.0**	43.6	42.2
400M	46.3	**46.8**	44.7	44.1
1.3B	**47.0**	46.8	46.4	44.7
7B	48.5	**48.6**	47.5	46.1
	ZS	**w/t Comet**	**Self-Talk**	
GPT2	41.1[5]	47.5	46.2	

Table 7: Zero-shot performance of Social IQa when using different knowledge bases. GPT2 results are taken from Shwartz et al. (2020). ZS: zero-shot performance; CN: ConceptNet.

[5]We achieved 46.7 when evaluating GPT2 in the zero-shot setting.

[6]We also experimented with other score functions, such as appending the extracted knowledge base triples to the question instead of the answer, although this approach does not yield better results than the one proposed by Shwartz et al. (2020).

formance of our well-tuned 1.3B zero-shot model—which does not benefit from any common-sense knowledge base triplets—nearly matches the performance of the GPT-2 model that is augmented with the Comet (Bosselut et al., 2019) knowledge base (47.0% for our zero-shot 1.3B model vs 47.5% for GPT-2 augmented with COMET; Table 7), and even outperforms the GPT-2 model that is augmented with self-talk (Shwartz et al., 2020). Nevertheless, we find that adding knowledge base triplets fails to yield substantial improvements for our models compared to those of (Shwartz et al., 2020); this finding is consistent across three different knowledge bases and five model sizes. On the contrary, adding common-sense knowledge base triplets can occasionally decrease performance compared to the zero-shot baseline.

We remark on two significant aspects of our findings. First, our findings highlight the importance of comparing proposed improvements against strong, well-tuned baselines (Henderson et al., 2018; Melis et al., 2018), which can achieve surprisingly competitive performance. In particular, we identify the choice of the scored span as an important evaluation hyper-parameter: While Shwartz et al. (2020) scored the GPT-2 model on the concatenation of both question and answer, we instead score the conditional probability of the answer given the question, which leads to better performance. Second, our findings highlight that certain improvements that are observed under a particular set of evaluation hyper-parameters may not necessarily be replicated under a different set of evaluation hyper-parameters. This finding further highlights the importance of explicitly stating the evaluation hyper-parameters used in each experiment, and identifying whether or not the improvements are robust across different evaluation hyper-parameters.

8 Related Work

While recent work evaluates large language models against commonsense benchmarks in a zero-shot way, they do not examine the extent to which these models exhibit commonsense understanding – do not consider strong baseline and simply compare their results with the state-of-the-art results produced under different experimental settings (Brown et al., 2020; Patwary et al., 2021).

Previous work has probed for commonsense knowledge in large language models using knowledge base completion (Petroni et al., 2019; Davison et al., 2019) or manually designed probing tasks (Weir et al., 2020; Shwartz and Choi, 2020). In contrast, we focus on the zero-shot performance of large language models on a range of commonsense benchmarks.

Another related line of work aims to improve the zero-shot performance of language models on commonsense benchmarks, either using external commonsense knowledge bases or the output generated by the language model (Bosselut et al., 2021; Ma et al., 2021; Shwartz et al., 2020; Paranjape et al., 2021; Holtzman et al., 2021).

Recent work (Trichelair et al., 2019; Elazar et al., 2021) also investigates the existence of dataset bias in two commonsense benchmarks, WinoGrande and SWAG(Zellers et al., 2019a). Finally, the work of (Zhou et al., 2020) is similar to our work in that they evaluate pre-trained language models against multiple commonsense benchmarks; they propose a new evaluation dataset that requires multi-hop reasoning. On the other hand, we focus on a systematic study using existing benchmarks and establishing guidelines for evaluating commonsense understanding in a zero-shot way.

9 Conclusion

We examine the extent to which large language models exhibit commonsense understanding by evaluating them against a range of commonsense benchmarks in a zero-shot way. At first sight, these models show impressive zero-shot performance suggesting that they capture commonsense knowledge. However, a closer inspection reveals that the good performance of these models is due to existing dataset bias in our benchmarks: the zero-shot performance is closer to that of a strong baseline that *does not* use questions at all than to the state-of-the-art performance. We also observe that as model size increases, the improvements in the zero-shot performance plateaus, indicating building larger models is not enough for achieving human-level commonsense understanding. In addition, the few-shot evaluation does not show notable improvements over the zero-shot setting except for Social IQa.

Our results suggest that the language modeling objective and larger model capacity are not enough to extract commonsense knowledge from text. Future work needs to explore alternative modeling paradigms to better capture commonsense knowledge from text. Moreover, given the implicit na-

ture of commonsense knowledge, the text modality alone might not be enough for improving the commonsense understanding capacity of NLP systems; an interesting future direction is leveraging other modalities such as images or videos to improve this capacity.

References

Lisa Bauer and Mohit Bansal. 2021. Identify, align, and integrate: Matching knowledge graphs to commonsense reasoning tasks. *EACL*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439.

Michael Boratko, Xiang Lorraine Li, Rajarshi Das, Tim O'Gorman, Dan Le, and Andrew McCallum. 2020. Protoqa: A question answering dataset for prototypical common-sense reasoning. *EMNLP 2020*.

Antoine Bosselut, Ronan Le Bras, and Yejin Choi. 2021. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *ACL*.

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

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. 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 7282–7296, Online. Association for Computational Linguistics.

Joe Davison, Joshua Feldman, and Alexander M Rush. 2019. Commonsense knowledge mining from pretrained models. 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 1173–1178.

Yanai Elazar, Hongming Zhang, Yoav Goldberg, and Dan Roth. 2021. Back to square one: Bias detection, training and commonsense disentanglement in the winograd schema. *arXiv preprint arXiv:2104.08161*.

John H Flavell. 2004. Theory-of-mind development: Retrospect and prospect. *Merrill-Palmer Quarterly (1982-)*, pages 274–290.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 25–30.

David Gunning. 2018. Machine common sense concept paper. *arXiv preprint arXiv:1810.07528*.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *Proc. of AAAI*.

Ari Holtzman, Peter West, Vered Schwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface

form competition: Why the highest probability answer isn't always right. *arXiv preprint arXiv:2104.08315*.

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

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *NMT@ACL*.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Joongbo Shin, and Kyomin Jung. 2021. Kpqa: A metric for generative question answering using keyphrase weights. *NAACL*.

Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William W Cohen. 2021. Differentiable open-ended commonsense reasoning. *NAACL*.

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

Hugo Liu and Push Singh. 2004. Commonsense reasoning in and over natural language. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 293–306. Springer.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *AAAI*.

John McCarthy et al. 1960. *Programs with common sense*. RLE and MIT computation center.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *Proc. of ICLR*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Prompting contrastive explanations for commonsense reasoning tasks. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Mostofa Patwary, Mohammad Shoeybi, Patrick LeGresley, Shrimai Prabhumoye, Jared Casper, Vijay Korthikanti, Vartika Singh, Julie Bernauer, Michael Houston, Bryan Catanzaro, Shaden Smith, Brandon Norick, Samyam Rajbhandari, Zhun Liu, George Zerveas, Elton Zhang, Reza Yazdani Aminabadi, Xia Song, Yuxiong He, Jeffrey Zhu, Jennifer Cruzan, Umesh Madan, Luis Vargas, and Saurabh Tiwary. 2021. Using deepspeed and megatron to train megatron-turing nlg 530b, the world's largest and most powerful generative language model.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? In *EMNLP*.

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

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on*

Artificial Intelligence, volume 34, pages 8732–8740.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019b. Socialiqa: Commonsense reasoning about social interactions. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Vered Shwartz and Yejin Choi. 2020. Do neural language models overcome reporting bias? In *COLING*.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, , and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *EMNLP*.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? 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 3354–3360, Hong Kong, China. Association for Computational Linguistics.

Paul Trichelair, Ali Emami, Adam Trischler, Kaheer Suleman, and Jackie Chi Kit Cheung. 2019. How reasonable are common-sense reasoning tasks: A case-study on the Winograd schema challenge and SWAG. 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 3382–3387, Hong Kong, China. Association for Computational Linguistics.

Trieu H Trinh and Quoc V Le. 2018. Do language models have common sense?

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Nathaniel Weir, Adam Poliak, and Benjamin Van Durme. 2020. Probing neural language models for human tacit assumptions. *arXiv: Computation and Language*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2019a. Swag: A large-scale adversarial dataset for grounded commonsense inference. *ACL*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019b. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9733–9740.

A Zero-shot Parameters

A.1 HellaSwag

Best

```
Score Function: Entropy
Prompt Format: [Question]+"question"+[Answer]:
Scored Text: Answer conditioned on question
```

Worst

```
Score Function: Sequence log probability
Prompt Format: [Question]+"question"+[Answer]:
Scored Text: Answer conditioned on question
```

A.2 Social IQa

Best

```
Score Function: Entropy
Prompt Format: Converted Text
Scored Text: Answer conditioned on question
```

Worst

```
Score Function: Sequence log probability
Prompt Format: [Question]+"question"+[Answer]:
Scored Text: Answer conditioned on question
```

Note: Independent experiment (use entropy (token-level log probability and converted text as the other two parameters) showed that if the scored-text is the concatenation of question and answer, the results could be worse.

A.3 WinoGrande

Best

```
Score Function: Sequence log probability
Prompt Format: Question with replaced pronoun with each answer choice
Scored Text: Question with replaced pronoun with each answer choice
```

Worst

```
Score Function: Entropy
Prompt Format: Question with replaced pronoun with each answer choice
Scored Text: Question with replaced pronoun with each answer choice
```

A.4 PIQA

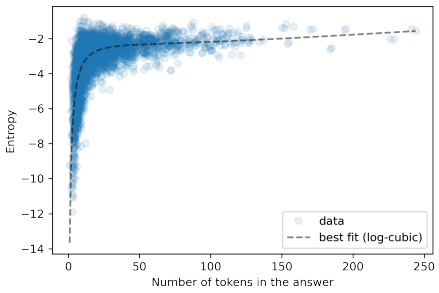
Best

```
Score Function: Entropy
Prompt Format: [Question]+"question"+[Answer]:
Scored Text: Answer conditioned on question
```

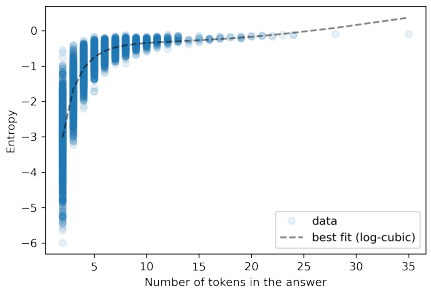
Worst

```
Score Function: PMI
Prompt Format: [Question]+"question"+[Answer]:
Scored Text: Answer conditioned on question
```

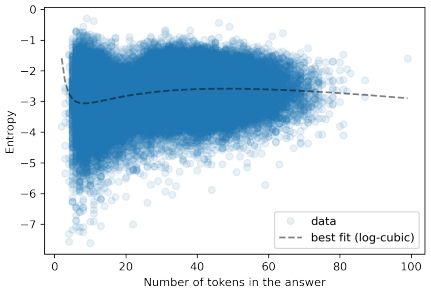
B Entropy vs answer length for all datasets



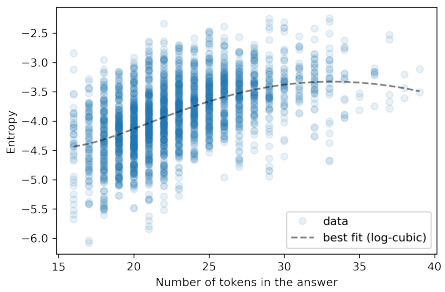
(a) Answer length vs entropy (average log probability across tokens) for PIQA.



(b) Answer length vs entropy (average log probability across tokens) for SocialIQA.



(a) Answer length vs entropy (average log probability across tokens) for HellaSWAG.



(b) Answer length vs entropy (average log probability across tokens) for Winogrande.