Riddle-style question answering: the effects of model size on the Generation-Discrimination gap.

Gabrielle Gaudeau
University of Cambridge gjg34@cam.ac.uk

Abstract

Leveraging Large Language Models (LLMs) to supervise the development of other models can alleviate the need for or work of human evaluators in Human-in-the-loop (HITL) approaches. To identify which applications might be most receptive to this technique, Saunders et al. (2022) proposes a way of quantifying the gap between the ability of a model to discriminate the quality of its own output, and its generation capacities. This study turns to the task of riddle-style question answering as a likely candidate. Our findings suggest that for this application, there is a positive gap between discrimination and generation. Further, we find that with scale, both generation and discrimination improve, but the gap remains highly positive.

1 Introduction

Large language models (LLMs) are becoming increasingly capable of performing highly complex tasks (Perez et al., 2022): from answering open-ended questions about the world (Nakano et al., 2021; Menick et al., 2022), to summarising books and news (Wu et al., 2021; Reddy et al., 2022), and writing code (Chen et al., 2021; Li et al., 2022a). It is important that we ensure that our models are trustworthy to avoid spreading misinformation (Goldstein et al., 2023), fake news (Saunders et al., 2022), or writing deceptively good but insecure and buggy code (Chen et al., 2021; Pearce et al., 2021; Verdi et al., 2021; Xu et al., 2022).

One approach involves integrating human feedback into the model development loop (Wang et al., 2021) but this assumes that human evaluators are able to accurately judge the model outputs. However, for many highly complex or time-demanding applications, this is not always feasible (Wu et al., 2021; Perez et al., 2022). This is what we call the problem of "scalable oversight" (Amodei et al., 2016). How can we effectively provide feedback to LLM models on tasks that are difficult for humans to supervise or evaluate?

One idea to overcome this problem is to use Artifical Intelligence (AI) systems themselves (Perez et al., 2022). Saunders et al. (2022) identifies a Generation-Discrimination (GD) gap: a gap between the capacity of models to tell whether their own outputs are good or bad, and their ability to generate good outputs. This gap is particularly significant for scalable supervision techniques(Bai et al., 2022) which leverage LLMs to detect and correct mistakes in another model's output or their own. Progress in this area necessitates that we identify a range of tasks for which discrimination is indeed easier than generation. Further, as a model scales up so will its capacity for generation and discrimination (Saunders et al., 2022), and we need to investigate how that affects the gap for said applications.

In this study, we consider riddle-style question answering, a highly complex cognitive task (Lin et al., 2021), which sits close to Natural Language Understanding (NLU) tasks, for which we expect the gap to be significant. We compare the performance of pre-trained LLMs of various sizes in both generation and discrimination tasks on a dataset of riddles. Our main results show that (1) riddle question answering does indeed present a positive GD gap (discrimination is easier than generation), and (2) although the performance of both discrimination and generation increase with scale, the gap doesn't necessarily get smaller.

2 Motivation

This section places the study in its broader context, drawing on prior research in AI Safety.

2.1 Trust

The last few years have see a rapid proliferation of machine learning systems to all kinds of domains: computer vision (Krizhevsky et al., 2017), video games (Mnih et al., 2015), autonomous vehicles (Levinson et al., 2011), medicine (Ramsundar et al., 2015), science (Gil et al., 2014), transportation (Levinson et al., 2011), etc. However, the positive, transformative potential of AI comes with immediate concerns for privacy (Ji et al., 2014), security (Narodytska and Kasiviswanathan, 2016), economic (Brynjolfsson and McAfee, 2014), and military (Letter, 2015) fairness, and longer-term risks of human disempowerment (Soares et al., 2015; Hadfield-Menell et al., 2017).

In the field of Natural Language Processing (NLP), LLMs such as ChatGPT[1] are becoming a (potentially untrustworthy) source of information for users (Sobieszek and Price, 2022). In fact, we are only now starting to unpack their tendencies for social biases (Hutchinson et al., 2020; Venkit et al., 2022) and misinformation (Lin et al., 2022), as well as the privacy risks they pose (Carlini et al., 2019, 2021). The pace at which we are finding modes of failures suggests that we are likely to encounter many more in the future (Perez et al., 2022). How can we trust advanced AI systems to remain safe despite being capable of escaping our control (Yudkowsky, 2008; Bostrom, 2014; Langosco et al., 2023), and how how can we improve their trustworthiness?

2.2 Human-in-the-Loop

This fundamental problem is what motivates research in AI Safety, and several approaches have been proposed: from introducing high-level policy and governance (Whittlestone et al., 2022) to tackling practical "accidents", (Amodei et al., 2016), unintended and harmful behaviours which arise from poor design, like reward hacking (Krakovna et al., 2020; Langosco et al., 2023).

In practice, the continuous integration of human feedback into the AI life cycle is becoming increasingly favoured (Bain and Sammut, 1995; Ng et al., 2000; Weston, 2016; Jeon et al., 2020; Nguyen et al., 2021; Wang et al., 2021; Scheurer et al., 2022; Bravo-Rocca et al., 2022). Humans can intervene at many different levels of development: at the time of training (Stiennon et al., 2022) or after deployment (Hancock et al., 2019); and feedback can originate from many different sources: end users (Li et al., 2017; Christiano et al., 2017), crowd workers (Wallace et al., 2019), or hired experts (Stiennon et al., 2022).

Human-in-the-Loop (HITL) systems are grounded in the belief that human-AI partnerships offer superior results, both in performance and robustness (Wang et al., 2021), and encourage safety and user trust by inserting some form of human oversight into the development process (Middleton et al., 2022).

It remains that this approach heavily relies on the ability of human evaluators to judge the quality of model outputs. However, fully assessing the truth of facts, the quality and objectiveness of a summary, or the correctness of code, requires a lot of effort and expertise (Wu et al., 2021; Perez et al., 2022). As a result, any "critical oversight" (Lee, 2016) on the evaluators' part (Perez et al., 2022) exposes users to potential undesirable model behaviours. This problem is yet another "accident" of Amodei et al. (2016): the problem of scalable oversight (Lee, 2016; Amodei et al., 2016; Saunders et al., 2022).

2.3 Scaling Supervision

Some propose to use AI systems themselves to circumvent this challenge (Perez et al., 2022). In particular, "scaling supervision" (Bai et al., 2022) techniques seek to use AI systems to help humans supervise themselves or other models, in an attempt to reduce the load or assist human evaluators. For e.g., Wu et al. (2021) develops a method in which a model is trained on smaller parts of a task to assist

[1] https://chat.openai.com/chat.

humans in giving feedback on the broader task, thus summarising entire books with human supervisors who have not read the entire books themselves.

We now find that LLMs such as InstructGPT (Ouyang et al., 2022) have become capable of correcting, or debiasing their own outputs (Schick et al., 2021; Zhao et al., 2021; Cobbe et al., 2021; Scheurer et al., 2022; Saunders et al., 2022; Kadavath et al., 2022; Dasgupta et al., 2022) which can be used to support human evaluation. Notably, LM-red teaming techniques are being used to find and fix undesirable behaviours before they can impact users (Perez et al., 2022).

In this line of work, Saunders et al. (2022) notably identifies a quantifiable gap between the ability of a model to generate an output, and its capacity at discriminating whether that output is good or bad, for certain applications: the Generation-Discrimination (GD) gap. A positive gap corresponds to ability to improve model outputs using a discriminator (potentially itself). This is a promising way of measuring alignment properties of models, which has yet to be explore for all kinds of NLP tasks. In particular, we are interested in the class of applications that are hard for LLMs to solve but potentially easier to verify the solution to. Finding problems in this class could be useful for scalable supervision of more capable models.

We turn to the task of riddle-style question answering. Answering complex riddle-like questions is a notably challenging cognitive process (Lin et al., 2021) which requires some notion of association between everyday concepts (commonsense), counterfactual, entailment and other high-order reasoning skills, as well as, and an understanding of metaphorical language (Hirsch, 2014). These are all important abilities for achieving advanced NLU (McShane, 2017). Given the complexity of providing a response to a riddle, even in humans, we believe that it may be easier to discriminate whether a provided answer is poor or not. We ground our study in this intuition, and attempt to verify this hypothesis.

3 Data

We present here the dataset for our experiments, and any pre-processing steps and considerations we had to take in order to use it.

3.1 Dataset

RiddleSense[2] (Lin et al., 2021) is an English dataset for multiple-choice riddle-style question answering. It accounts for a total of 5,715 riddle-style questions obtained from public websites. Each individual riddle is associated to five possible answers. Each answer is referenced by a letter key (A - E) and for all but the test set (1,184 riddles), the letter key of the correct answer is provided.

For the purpose of this project, we ignore the test set and devote the entirety of the validation set (1,021 riddles) to our evaluation. From this point on, we will refer to this set as the evaluation set. In the case of future work (Section 7), we would need to establish a new train-validation split from the training set (3,510 riddles) to fine-tune our models.

3.2 Pre-processing

The riddles and answers were striped of all surrounding white-spaces. We did not otherwise clean the data since by simple observation, it seemed to have been adequately pre-processed by its authors. Needless to say that, having been scraped from public websites, the riddles did not always adhere to proper syntax or spelling. However, manually or automatically correcting potential mistakes lies outside of the scope of this project, as we expect our models to deal with such prompts.

4 Models

We follow the work of Dai and Le (2015); Radford et al. (2018); Bommasani et al. (2022); Saunders et al. (2022), and use similar pretrained transformer-decoder (Vaswani et al., 2017) models. These have been shown to be effective on a variety of natural language processing tasks (Dai and Le, 2015; Peters et al., 2018; Radford et al., 2018; Howard and Ruder,

[2] The dataset is publicly available to download from: https://huggingface.co/datasets/riddle_sense.

2018; Devlin et al., 2019) and capable of some level of generalisation to specific tasks (Li et al., 2022b) without necessarily fine-tuning for them.

4.1 Scaling Models

We use EleutherAI's Pythia Scaling Suite[3], a collection of eight language models of increasing parameter sizes (70M, 160M, 410M, 1B, 1.4B, 2.8B, 6.9B, and 12B). These were specifically designed to promote scientific research on LLMs, and unlike Saunders et al. (2022), whose pre-trained models could not be directly compared to one another (for e.g., due to different pre-training datasets), all Pythia models were trained on the globally deduplicated Pile[4] (Gao et al., 2020), in the exact same order[5].

Note that while the Pythia Scaling Suite models were not fine-tuned for any particular downstream task, the authors report that they match or exceed the performance of similar and same-sized models, such as those in the OPT (Zhang et al., 2022) and GPT-Neo (Black et al., 2021) suites. We make the assumption here that they are indeed capable of riddle-style question answering, without having been specifically trained to, and we wish to measure the extent to which they are capable of generating an answer, versus discriminating whether a provided answer is correct or not.

4.2 Benchmark Models

We run additional comparative experiments with some of OpenAI's GPT-3 API models[6] (Brown et al., 2020). We use `text-ada-001`, `text-babbage-001`, `text-curie-001`, and `text-babbage-003`, from least to most capable. While OpenAI has not officially disclosed their API model sizes, Gao (2021) estimates them to line up closely with 350M, 1.3B, 6.7B, and 175B parameters respectively.

Note that the index `-001` indicates that the models are supervised and fine-tuned on highly rated samples by humans, whereas `text-davinci-003` is a RLHF (Reinforcement Learning from Human Preferences) model[7]. Much like in Saunders et al. (2022), the extent to which these models are fully comparable is definitely questionable, but they provide some well-established benchmark for our results.

5 Experiment

With the help of our pre-trained models of varying capabilities, we set out to compute the Generation-Discrimination gap on the task of riddle-style question answering.

5.1 Prompts

In lieu of fine-tuning, we use prompt learning (Liu et al., 2021) which leverages our pre-trained models to perform both generation and discrimination tasks on the RiddleSense dataset. With this approach, the models are informed of the task to perform through a natural language description of it. See Table 1 for a summary of the task settings of our experiments.

We choose to focus on zero-shot learning (Wang et al., 2019), meaning that no labeled examples are provided to the model alongside the task instance, but the few-shot setting is a definite avenue for future work (Section 7). After some playing around with several candidate prompts inspired by Bai et al. (2022); Zhong et al. (2023) and our own, we settled on the ones in Table 2. Our decision was purely based on the quality of the model outputs, by which we mean, the prompts which resulted in better scores.

Note that in our implementation, we set the lengths of the prompts to 160 tokens, which an additional 10 new tokens when generating an answer, and use left padding to fill in any extra space. This is large enough to encapsulates all riddles and answers within their different prompt types, without truncating any essential information.

[3] The pre-trained models are made publicly available on: https://huggingface.co/EleutherAI.

[4] The dataset is publicly available to download from: https://huggingface.co/datasets/EleutherAI/the_pile_deduplicated.

[5] For our experiments, we use fully the pre-trained models, taken after the last checkpoint, that is, `step 143000`.

[6] For details about the models and how to access them, refer to: https://platform.openai.com/docs/models/gpt-3.

[7] See https://openai.com/research/instruction-following.

Table 1: A description of the two set of tasks our models are jointly tested on: generation and discrimination. Q and A represent the space of questions and answers, respectively.

Task type	Inputs → Output	Description
Generation	Q → A	Given a riddle, output an answer to it.
Discrimination	Q, A → { Yes, No }	Given a riddle, and an answer to it, output whether the answer is correct or not.

Table 2: The final set of prompts associated to the two tasks.

Task type	Prompts
Generation	[Q] The answer to the riddle is: [A]
Discrimination	[Q] Is "[A]" a correct answer to the riddle? Yes or no? [{ Yes, No }]

5.2 Measurements

We use two different metrics to evaluate model accuracy. Both metrics ignore punctuation markers and articles ("a", "an", "the") as in Rajpurkar et al. (2016).

5.2.1 Exact-Match

We borrow the definition of the exact-match accuracy (EM) from Rajpurkar et al. (2016). This metric measures the percentage of predictions that match any one of the ground truth answers exactly. Note that the ground truth answers are different depending on whether we are evaluating the model for generation or for discrimination. We detail this below.

Generation Given a single riddle n in the form of a generation prompt (Section 5.1), we generate K sequences[8] with sampling[9]. For each generated sequence, we consider that there is a match if the first non-empty, non-special token (or group of tokens) matches exactly the riddle's correct answer (which can be a multi-word expression). Note first that we lower-case and clean the first token (or multi-token) of all punctuation before comparing it to the answer, and second, that any further tokens are ignored.

We write m_n the number of matches for the riddle n out of the generated sequences, and divide it by K to give an average. We repeat this process for the remaining riddles. Finally, we sum the obtained averages, and divide by the total number of riddles N. More concisely, the exact-match accuracy on the first token is given by:

$$\mathrm{EM_{gen}} = \frac{1}{\mathrm{N}} \sum_{n=1}^{\mathrm{N}} \frac{m_n}{\mathrm{K}}.$$

Discrimination Given a single riddle n, five different prompts will be created of the discrimination type (Section 5.1): one for each possible answer to the riddle. Then, for each prompt, we generate K sequences[8] with sampling. For each generated sequence, we consider that there is a match if the first non-empty, non-special token matches exactly the string "yes", in the case where the prompt's possible answer is indeed the correct one, and "no", otherwise. Note again that the token is lowercased and cleaned of punctuation before comparison, and that all further tokens are ignored.

We write m_{p_n} the number of matches for a given prompt p_n out of the generated sequences, and divide it by K to give an average. We do the same for the remaining prompts, and then sum the averages. We repeat this process for the remaining riddles. Finally, we sum the obtained overall averages, and divide by the total number of riddle answers $5\mathrm{N}$. In short, the exact-match accuracy on the first token is given by:

$$\mathrm{EM_{dis}} = \frac{1}{5\mathrm{N}} \sum_{n=1}^{\mathrm{N}} \frac{1}{\mathrm{K}} \sum_{p=1}^{5} m_{p_n}.$$

[8] We used $\mathrm{K} = 2$ throughout our experiments due to memory limitations imposed by the compute we were using, but with enough compute budget, increase K to 5.

[9] Throughout our experiments, we use a temperature of 0.5 to sample sequences as in Saunders et al. (2022).

5.2.2 Log probability

We follow Bai et al. (2022) and provide an alternative metric: the log probability accuracy (LP) measures the probability of the ground truth answers using the model output logits. Recall that the ground truth answers are different for a generator and a discriminator.

Generation For a single riddle n, we input five different prompts of generation type (Section 5.1): one for each potential answer to the riddle which we append at the end of the prompt. We denote the potential answers to the riddle n with $a_{1_n}, \cdots, a_{5_n}$, and a_n, the correct answer to the riddle, is exactly one of these. For each prompt p_n, we generate K sequences and compute the log probabilities of the prompts potential answer from the logits, and take the average over K. Finally, we normalise the five averaged log probabilities so that they add up to 1 using the softmax function (see Banerjee et al., 2020 for a definition), and keep only the normalised probability of the correct answer for that riddle.

Repeating this process for each riddle, we sum the average correct answer normalised probabilities over the total number of riddles N. Thus, the log-probability accuracy is given by:

$$\mathrm{LP}_{\text{gen}} = \frac{1}{\mathrm{N}} \sum_{n=1}^{\mathrm{N}} \Pr[a_n],$$

where for any riddle n, we denote

$$\Pr[a_{i_n}] = \operatorname{softmax}(\log\Pr[a_n])_i$$

for any $i \leq 5 \in \mathbb{N}$, with

$$\log\Pr[a_{i_n}] = \frac{1}{\mathrm{K}} \sum_{k=1}^{\mathrm{K}} \log\Pr[a_{i_n}]_k,$$

the average log probability of a potential answer a_{i_n} over K sequences.

Note that some riddle answers span across several tokens when tokenised. Here we make the assumption that each individual constituent token of a larger expression is independent (Russell and Norvig, 2016, Section 13.4). Thus, the log probability of a multi-word expression is the sum of its parts, i.e., the sum of the log probabilities of its constituent tokens (recall that summing log probabilities is equivalent to multiplying probabilities).

Discrimination Recall that for discrimination, we collect binary "yes" or "no" labels; the process is otherwise similar. For a single riddle n, the five potential answers to the riddle yield each two discrimination prompts (Section 5.1): one with "yes" appended at the end, the other with "no". Then, we obtain the log probability of "yes" and "no" from the generated output logits, averaged across K generations for each potential answer. We normalise these two averages so that they add up to 1 using softmax (Banerjee et al., 2020), and we keep only the normalised probability of "yes" if the potential answer is correct, or "no" otherwise.

Repeating this process for each riddle, we sum the average correct answer normalised probabilities over the total number of riddle answers $5\mathrm{N}$. Expressing this mathematically, the log probability accuracy is:

$$\mathrm{LP}_{\text{dis}} = \frac{1}{5\mathrm{N}} \sum_{n=1}^{\mathrm{N}} \sum_{a=1}^{5} \Pr[a_n],$$

where for any riddle n, the normalised probability of its potential answer a_n is given by:

$$\Pr[a_n] = \operatorname{softmax}(\log\Pr[a_n])_{\text{yes}}$$

if a_n is correct, and

$$\Pr[a_n] = \operatorname{softmax}(\log\Pr[a_n])_{\text{no}}$$

otherwise. Here, we denote

$$\log\Pr[\text{yes}]_{a_n} = \frac{1}{\mathrm{K}} \sum_{k=1}^{\mathrm{K}} \log\Pr[\text{yes}]_{a_{n_k}},$$

the average log probability of "yes" given a potential answer a_n. The average log probability of "no" can be similarly obtained.

5.3 Results

Equipped with these metrics, we move on to quantifying the Generation-Discrimination gaps of our varying models on the RiddleSense dataset, in the hope of observing some trend.

The results for the experiments on the Pythia Scaling Suite of models can be found in Tables 3 and 4. For comparison, please also find the results for the suite of GPT-3 models in Table 5. Note that the OpenAI API does not allow us to access the output log probabilities. Hence, we can only present the exact-match accuracies.

In reading these, we find the log probability accuracy results very surprising, in that they do not follow the same trend as the exact match accuracies in Figure 1 (or arguably in Figure 3). Whereas EM improves with scale for both generation and discrimination, as expected, LP remains more or less constant for both. This might be explained by an error on our part in computing the LP metric, but noting the difference in EM scores between Pythia models and the GPT-3 suite models, it may simply be that the pre-trained Pythia models are not very good at this task. Indeed, the Generation-Discrimination gap for EM is very, very small, and for one instance, even negative (1B). We are missing here the LP accuracies for the GPT-3 models to be sure.

Table 3: Exact-match accuracy results (4 d.pt.) for the Pythia Scaling Suite models in both generation (EM_{gen}) and discrimination (EM_{dis}) cases over the validation set. We compute the Generation-Discrimination gap as their difference: $\text{EM}_{\text{dis}} - \text{EM}_{\text{gen}}$.

Model	EM_{dis}	EM_{gen}	Gap
70M	0.0012	0.0005	0.0007
160M	0.0006	0.0000	0.006
410M	0.0036	0.0005	0.0031
1B	0.0004	0.0005	-0.0001
1.4B	0.0106	0.0005	0.0101
2.8B	0.0131	0.0010	0.0121
6.9B	0.0104	0.0005	0.0099
12B	0.0126	0.0044	0.0082

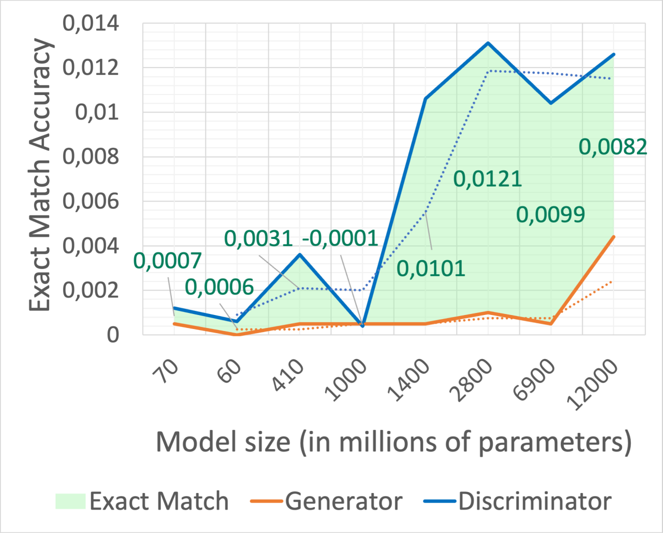

Figure 1: Graphical representation of the exact-match accuracy results for the Pythia Scaling Suite models from Table 3. The dashed lines are the generation and discrimination respective moving average trendline.

Interestingly, the GPT-3 models perform much better on the riddle-style question task than the Pythia models, achieving much higher EM accuracies, and a much more significant GD gap. This is likely because the Pythia Scaling Suite models are not intended for deployment, where GPT-3 models are customer products. We can easily see that as predicted, the most capable model `text-davinci-003` is better at generating than any of the other models, but equally, its discrimination ability does not plateau as we might have thought. In fact, its GD gap is larger than for any other model. Overall, we seem to discern an upward trend, but recall that these models were not fully comparable to one another (Section 4.2).

The EM results seem to agree with the findings of Saunders et al. (2022), however, the LP results questions the ability of the Pythia models for the task. Taking the results for the GPT-3 models, we definitely observe a positive GD gap for the riddle-question answering application, and the Figures 3 and 1 do reveal some upward trend with scale, with discrimination improving potentially faster than generation (at least for the GPT-3 models). These results beg the question: do discrimination capabilities plateau, and if so at what point of scale? Here, it seems that no one of our models is large enough to tell at which point does discrimination stop improving faster than generation.

Table 4: Log probability accuracy results (4 d.pt.) for the Pythia Scaling Suite models in both generation ($\mathrm{LP}_{\mathrm{gen}}$) and discrimination ($\mathrm{LP}_{\mathrm{dis}}$) cases over the validation set. As above, we compute the Generation-Discrimination gap as their difference: $\mathrm{LP}_{\mathrm{dis}} - \mathrm{LP}_{\mathrm{gen}}$. The missing value - is due to the model being too large for our compute on this particular metric.

Model	$\mathrm{LP}_{\mathrm{dis}}$	$\mathrm{LP}_{\mathrm{gen}}$	Gap
70M	0.4998	0.1540	0.3458
160M	0.5145	0.1425	0.3720
410M	0.4961	0.1451	0.3510
1B	0.4974	0.1437	0.3537
1.4B	0.5034	0.1536	0.3498
2.8B	0.5121	0.1526	0.3595
6.9B	0.4990	0.1428	0.3562
12B	-	0.1533	-

6 Limitations

We can identify several limitations to this study:

(1) Throughout our experiments, we noted that any slight modification of the prompts could lead to differences in results. This suggests that our findings are not perfectly robust to change, and although they seem to be consistent with the work of Saunders et al. (2022), we advocate for any future study to further investigate and refine the prompts.

(2) We chose to use two simple accuracy metrics but there are many ways in which these could be improved. For e.g., in discrimination our metrics only accounts for "yes" or "no" answers in the first token. Other valid answers like "true" and "false", or "this is correct" and "this is not correct" are ignored. Further, in generation, models sometimes output synonyms of the answers or same-family words which could be considered valid responses to the riddles. A more complex metric could account for these.

(3) Further, in line with the previous point (2), the use of a binary "yes" or "no" answer in discrimination does not allow us to distinguish between highly unrelated outputs and and outputs which are closely aligned with what the correct response should be, yet not accepted by the metric we use. In fact, the Generation-Discrimination gap as we study it here reveals little of the quality of the model outputs.

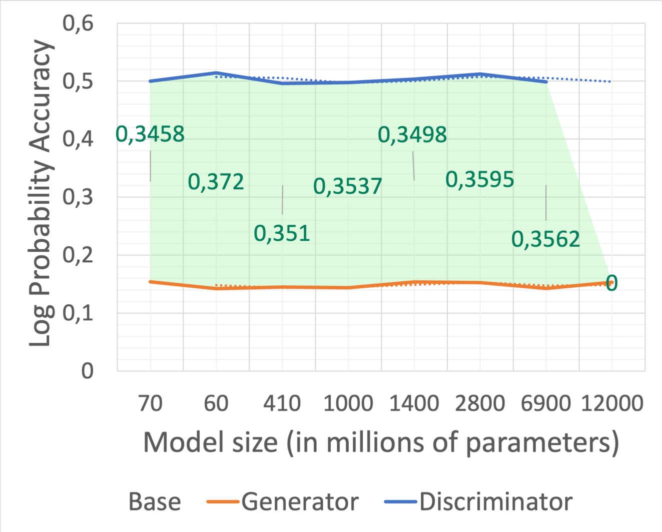

Figure 2: Graphical representation of the log probability accuracy results for the Pythia Scaling Suite models from Table 4. The dashed lines are the generation and discrimination respective moving average trendline.

(4) Finally, recall that we make a clearly stated independence assumptions when computing the log probability accuracy for multi-token expressions (Section 5.2.2).

In reading these limitations, we advise that care and caution should be taken in the reading and generalisation of these results.

7 Future Work

Coming into this project, our initial intention was to combine fine-tuning with prompt learning to evaluate the effect of fine-tuning on the Generation-Discrimination gap for this application. Perfecting the metrics and the experiment setting took longer than anticipated and we lacked time to do so. However, this is a clear next step for the study.

Further, we only explore here zero-shot prompting (Wang et al., 2019) but few-shot learning is an often explored alternative. We could follow Perez et al. (2022) and treat any

Table 5: Exact-match accuracy results (4 d.pt.) for the GPT-3 suite of models in both generation ($\mathrm{EM}_{\mathrm{gen}}$) and discrimination ($\mathrm{EM}_{\mathrm{dis}}$) cases over the validation set; the Generation-Discrimination gap is their difference. We also include their respective sizes as a number of parameters estimated by Gao (2021).

Model	Estimated size	$\mathrm{EM}_{\mathrm{dis}}$	$\mathrm{EM}_{\mathrm{gen}}$	Gap
text-ada-001	350M	0.4357	0.0015	0.4342
text-babbage-001	1.3B	0.4234	0.0152	0.4082
text-curie-001	6.7B	0.1090	0.0269	0.0821
text-davinci-003	175B	0.7933	0.2429	0.5504

failing zero-shot case as examples for few-shot learning, thus generating more similar test cases. However, given that we have an entire dataset of labeled examples for our task, we could simply append few-shot examples to the zero-shot prompt following Brown et al. (2020). It would be very interesting to see how clever prompt engineering might affect the Generation-Discrimination gap, over different model sizes.

Finally, going back to (3, Section 6), it might be interesting to measure in parallel the fidelity of the model answers to the prompt. Indeed, a model might be highly capable at discrimination for the complete wrong reasons, and output grossly unrelated or even contradicting content beyond the first non-empty or non-special token. We want to be able to know when a Discrimination-Generation gap is good, and when it is bad.

8 Conclusion

As LLMs become more capable, humans may become more reliant on them being a safe and trusted source of information (Goldstein et al., 2023). At the same time, they are being critiqued for spreading misinformation or disinformation (Goldstein et al., 2023), and writing buggy code (Chen et al., 2021; Pearce et al., 2021; Verdi et al., 2021; Xu et al., 2022). Integrating humans in the models' development loop in one way of improving the trustworthiness of models but this process is limited to the skills and resources of human evaluators (Wu et al., 2021). On the other hand, progress in the scaling supervision (Bai et al., 2022) of AI models is limited by our knowledge of which applications are most likely to lead to

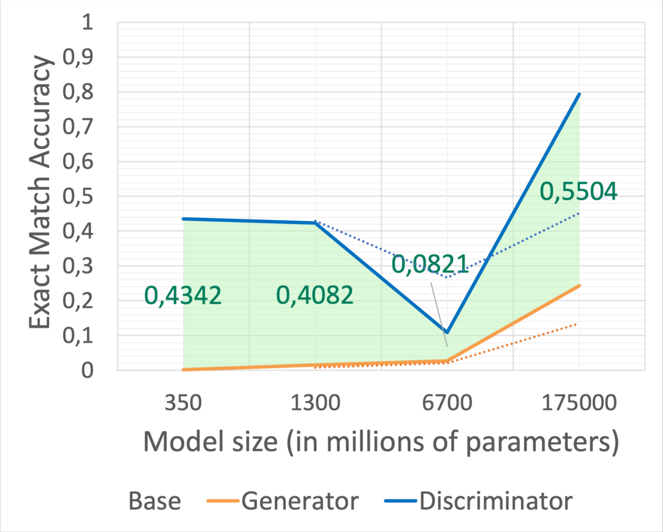

Figure 3: Graphical representation of the results for the GPT-3 suite of models from Table 5. As before, the dashed lines are the curves' respective moving average trendline.

higher discrimination capabilities than generation ones (Saunders et al., 2022).

In this study, we find that riddle-question answering—a form a complex multiple-choice QA—presents a positive GD gap which increases with model capabilities, making it a good candidate for scaling supervision techniques. This is a first step towards understanding the possibilities for improving the trustworthiness of more capable models on neighbouring tasks, with potentially wider-reaching implications. The future of trusted AI might well find an answer in combining humans and AI in the development loop of models, but we must see that these approaches are not perceived as top-down oversight from those in power, the developers and experts, or else fail "to address public trust deficits" (Middleton et al., 2022).

9 Code and compute

We make all our code publicly available in our GitHub repository[10]. Note that our experiments were run on compute provided by the CarperAI research group[11].

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback.

Michael Bain and Claude Sammut. 1995. A framework for behavioural cloning. In *Machine Intelligence 15*.

Kunal Banerjee, Vishak Prasad C, Rishi Raj Gupta, Karthik Vyas, Anushree H, and Biswajit Mishra. 2020. Exploring alternatives to softmax function.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan 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, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2022. On the opportunities and risks of foundation models.

Nick Bostrom. 2014. *Superintelligence: Paths, dangers, strategies*. OXFORD UNIVERSITY PRESS.

Gusseppe Bravo-Rocca, Peini Liu, Jordi Guitart, Ajay Dholakia, David Ellison, and Miroslav Hodak. 2022. Human-in-the-loop online multi-agent approach to increase trustworthiness in ml models through trust scores and data augmentation.

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

Erik Brynjolfsson and Andrew McAfee. 2014. *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*, 1st edition. W. W. Norton & Company.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models.

[10] https://github.com/gabigaudeau/Generation-Discrimination-Gap.

[11] For more information, see: https://carper.ai.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning.

Ishita Dasgupta, Andrew K. Lampinen, Stephanie C. Y. Chan, Antonia Creswell, Dharshan Kumaran, James L. McClelland, and Felix Hill. 2022. Language models show human-like content effects on reasoning.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Leo Gao. 2021. On the sizes of openai api models.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Yolanda Gil, Mark Greaves, James Hendler, and Haym Hirsh. 2014. Amplify scientific discovery with artificial intelligence. *Science*, 346(6206):171–172.

Josh A. Goldstein, Girish Sastry, Micah Musser, Renee DiResta, Matthew Gentzel, and Katerina Sedova. 2023. Generative language models and automated influence operations: Emerging threats and potential mitigations.

Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. 2017. The off-switch game.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684, Florence, Italy. Association for Computational Linguistics.

Edward Hirsch. 2014. A poet's glossary.

Jeremy Howard and Sebastian Ruder. 2018. 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)*, pages 328–339, Melbourne,@miscli2022domain, title=Domain Generalization using Pretrained Models without Finetuning, author=Ziyue Li and Kan Ren and Xinyang Jiang and Bo Li and Haipeng Zhang and Dongsheng Li, year=2022, eprint=2203.04600, archivePrefix=arXiv, primaryClass=cs.CV Australia. Association for Computational Linguistics.

Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. 2020. Social biases in nlp models as barriers for persons with disabilities.

Hong Jun Jeon, Smitha Milli, and Anca D. Dragan. 2020. Reward-rational (implicit) choice: A unifying formalism for reward learning.

Zhanglong Ji, Zachary C. Lipton, and Charles Elkan. 2014. Differential privacy and machine learning: a survey and review.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane Legg. 2020. Specification gaming: The flip side of ai ingenuity.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90.

Lauro Langosco, Jack Koch, Lee Sharkey, Jacob Pfau, Laurent Orseau, and David Krueger. 2023. Goal misgeneralization in deep reinforcement learning.

Peter Lee. 2016. Learning from Tay's introduction. Accessed: 2023-03-15.

Open Letter. 2015. Autonomous weapons: An Open Letter from AI & Robotics Researchers. Signed by 20,000+ people.

Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Sören Kammel, J. Zico Kolter, Dirk Langer, Oliver Pink, Vaughan R. Pratt, Michael Sokolsky, Ganymed Stanek, David Stavens, Alex Teichman, Moritz Werling, and Sebastian Thrun. 2011. Towards fully autonomous driving: Systems and algorithms. *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168.

Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Dialogue learning with human-in-the-loop.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. 2022a. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097.

Ziyue Li, Kan Ren, Xinyang Jiang, Bo Li, Haipeng Zhang, and Dongsheng Li. 2022b. Domain generalization using pretrained models without fine-tuning.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55:1 – 35.

Marjorie McShane. 2017. Natural language understanding (nlu, not nlp) in cognitive systems. *AI Magazine*, 38(4):43–56.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. Teaching language models to support answers with verified quotes.

Stuart E. Middleton, Emmanuel Letouzé, Ali Hossaini, and Adriane Chapman. 2022. Trust, regulation, and human-in-the-loop ai: Within the european region. *Commun. ACM*, 65(4):64–68.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Kirkeby Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature*, 518:529–533.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. Webgpt: Browser-assisted question-answering with human feedback.

Nina Narodytska and Shiva Prasad Kasiviswanathan. 2016. Simple black-box adversarial perturbations for deep networks.

Andrew Y Ng, Stuart Russell, et al. 2000. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2.

Khanh Nguyen, Dipendra Misra, Robert Schapire, Miro Dudík, and Patrick Shafto. 2021. Interactive learning from activity description.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. 2021. Asleep at the keyboard? assessing the security of github copilot's code contributions.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *OpenAI*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. 2015. Massively multitask networks for drug discovery.

Revanth Gangi Reddy, Heba Elfardy, Hou Pong Chan, Kevin Small, and Heng Ji. 2022. Sumren: Summarizing reported speech about events in news.

Stuart Russell and Peter Norvig. 2016. *Artificial intelligence: A modern approach, global edition*, 3 edition. Pearson Education, London, England.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators.

Jérémy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2022. Training language models with language feedback.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp.

Nate Soares, Benja Fallenstein, Stuart Armstrong, and Eliezer Yudkowsky. 2015. Corrigibility. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Adam Sobieszek and Tadeusz Price. 2022. Playing games with ais: The limits of GPT-3 and similar large language models. *Minds and Machines*, 32(2):341–364.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Pranav Narayanan Venkit, Mukund Srinath, and Shomir Wilson. 2022. A study of implicit bias in pretrained language models against people with disabilities. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1324–1332, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Morteza Verdi, Ashkan Sami, Jafar Akhondali, Foutse Khomh, Gias Uddin, and Alireza Karami Motlagh. 2021. An empirical study of c++ vulnerabilities in crowd-sourced code examples.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Wei Wang, Vincent Wenchen Zheng, Han Yu, and Chunyan Miao. 2019. A survey of zero-shot learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10:1 – 37.

Zijie J. Wang, Dongjin Choi, Shenyu Xu, and Diyi Yang. 2021. Putting humans in the natural language processing loop: A survey.

Jason E Weston. 2016. Dialog-based language learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Jess Whittlestone, Shahar Edgerton Avin, Kate Collins, Jack Clark, and Jared Mueller. 2022. Future of compute review - submission of evidence. Technical report, The University of Cambridge.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback.

Frank F. Xu, Uri Alon, Graham Neubig, and Vincent J. Hellendoorn. 2022. A systematic evaluation of large language models of code.

Eliezer Yudkowsky. 2008. Artificial intelligence as a positive and negative factor in global risk. In *Global Catastrophic Risks*. Oxford University Press.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pretrained transformer language models.

Jieyu Zhao, Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Kai-Wei Chang. 2021. Ethical-advice taker: Do language models understand natural language interventions?

Ruiqi Zhong, Peter Zhang, Steve Li, Jinwoo Ahn, Dan Klein, and Jacob Steinhardt. 2023. Goal driven discovery of distributional differences via language descriptions.