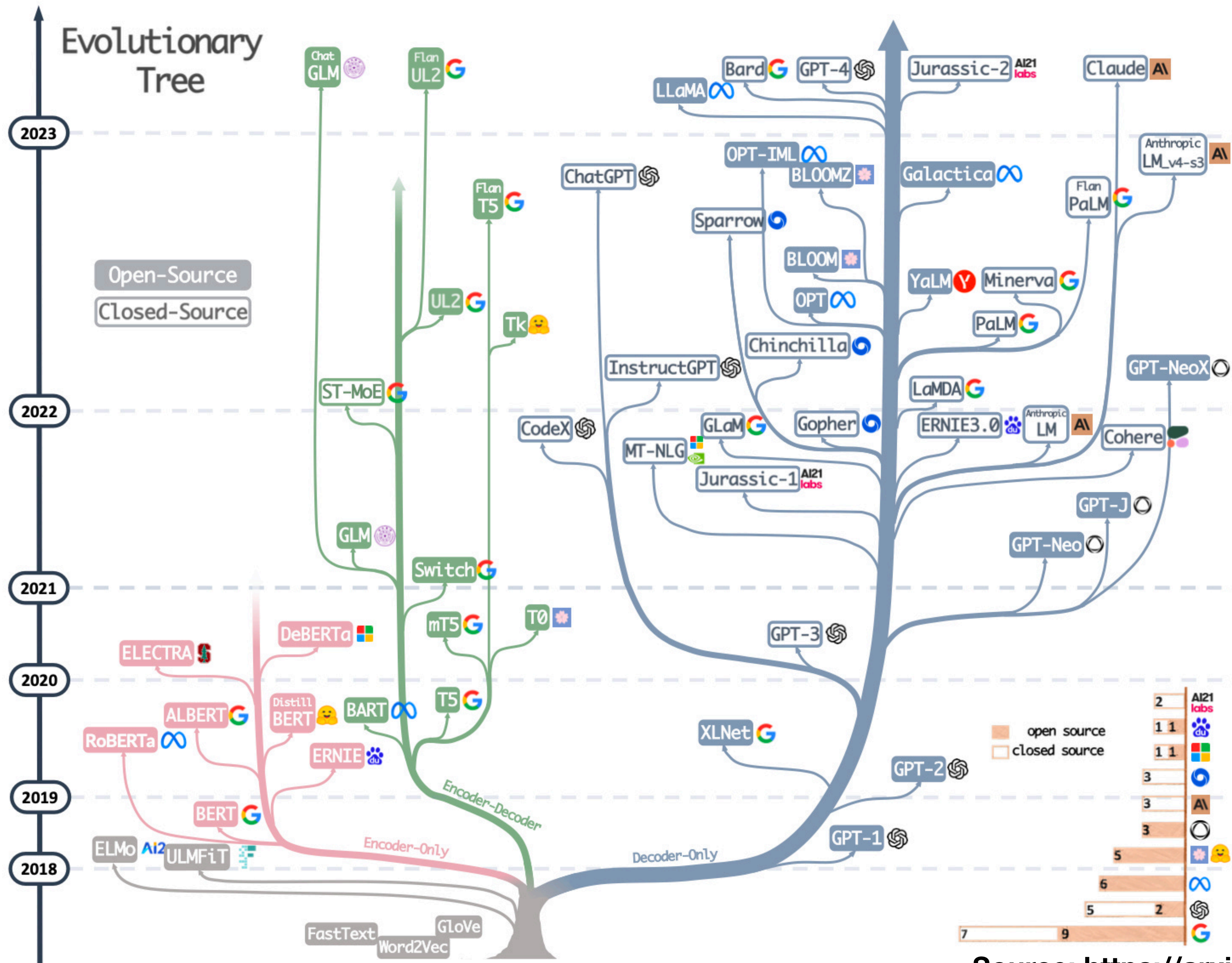


GPT-style language models, generalization bounds

493 / 599 May 15 2023

Ludwig Schmidt



Source: <https://arxiv.org/abs/2304.13712>

LLaMA: Open and Efficient Foundation Language Models

**Hugo Touvron*, Thibaut Lavril*, Gautier Izacard*, Xavier Martinet
Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Naman Goyal
Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin
Edouard Grave*, Guillaume Lample***

Meta AI

Abstract

We introduce LLaMA, a collection of foundation language models ranging from 7B to 65B parameters. We train our models on trillions of tokens, and show that it is possible to train state-of-the-art models using publicly available datasets exclusively, without resorting to proprietary and inaccessible datasets. In particular, LLaMA-13B outperforms GPT-3 (175B) on most benchmarks, and LLaMA-65B is competitive with the best models, Chinchilla-70B and PaLM-540B. We release all our models to the research community¹.

performance, a smaller one trained longer will ultimately be cheaper at inference. For instance, although [Hoffmann et al. \(2022\)](#) recommends training a 10B model on 200B tokens, we find that the performance of a 7B model continues to improve even after 1T tokens.

The focus of this work is to train a series of language models that achieve the best possible performance at various inference budgets, by training on more tokens than what is typically used. The resulting models, called *LLaMA*, ranges from 7B to 65B parameters with competitive performance

Model

| params | dimension | n heads | n layers | learning rate | batch size | n tokens |
|--------|-----------|-----------|------------|---------------|------------|------------|
| 6.7B | 4096 | 32 | 32 | $3.0e^{-4}$ | 4M | 1.0T |
| 13.0B | 5120 | 40 | 40 | $3.0e^{-4}$ | 4M | 1.0T |
| 32.5B | 6656 | 52 | 60 | $1.5e^{-4}$ | 4M | 1.4T |
| 65.2B | 8192 | 64 | 80 | $1.5e^{-4}$ | 4M | 1.4T |

Table 2: **Model sizes, architectures, and optimization hyper-parameters.**

| | GPU Type | GPU Power consumption | GPU-hours | Total power consumption | Carbon emitted (tCO ₂ eq) |
|------------|-----------|-----------------------|-----------|-------------------------|--------------------------------------|
| OPT-175B | A100-80GB | 400W | 809,472 | 356 MWh | 137 |
| BLOOM-175B | A100-80GB | 400W | 1,082,880 | 475 MWh | 183 |
| LLaMA-7B | A100-80GB | 400W | 82,432 | 36 MWh | 14 |
| LLaMA-13B | A100-80GB | 400W | 135,168 | 59 MWh | 23 |
| LLaMA-33B | A100-80GB | 400W | 530,432 | 233 MWh | 90 |
| LLaMA-65B | A100-80GB | 400W | 1,022,362 | 449 MWh | 173 |

Table 15: **Carbon footprint of training different models in the same data center.** We follow [Wu et al. \(2022\)](#) to compute carbon emission of training OPT, BLOOM and our models in the same data center. For the power consumption of a A100-80GB, we take the thermal design power for NVLink systems, that is 400W. We take a PUE of 1.1 and a carbon intensity factor set at the national US average of 0.385 kg CO₂e per KWh.

| Dataset | Sampling prop. | Epochs | Disk size |
|---------------|----------------|--------|-----------|
| CommonCrawl | 67.0% | 1.10 | 3.3 TB |
| C4 | 15.0% | 1.06 | 783 GB |
| Github | 4.5% | 0.64 | 328 GB |
| Wikipedia | 4.5% | 2.45 | 83 GB |
| Books | 4.5% | | |
| ArXiv | 2.5% | | |
| StackExchange | 2.0% | | |

Table 1: **Pre-training data.** Data m...
training, for each subset we list th...
tion, number of epochs performed...
training on 1.4T tokens, and disk si...
runs on 1T tokens have the same sa...

shawn swyx wang
@swyx

LLM datasets be like:

- First you start with CommonCrawl
- Then you add C4, which is just CommonCrawl again, but dont worry about it
- Eye of newt and Toe of frog
- Then Wikipedia (also is in CommonCrawl but dw)
- GitHub (permissively licensed, trust us)
- Wool of bat and tongue of dog
- Books. just all the Books.
- Every Arxiv paper
- Adder's fork and blind worm's sting
- StackExchange Q&As
- Lizard's leg and howler's wing

8:54 PM · May 2023 · 54.2K Views

73 Retweets 6 Quotes 623 Likes 89 Bookmarks

C4 (Colossal Clean Crawled Corpus)

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

| | |
|----------------|-------------------------|
| Colin Raffel* | CRAFFEL@GMAIL.COM |
| Noam Shazeer* | NOAM@GOOGLE.COM |
| Adam Roberts* | ADAROB@GOOGLE.COM |
| Katherine Lee* | KATHERINELEE@GOOGLE.COM |
| Sharan Narang | SHARANNARANG@GOOGLE.COM |
| Michael Matena | MMATENA@GOOGLE.COM |
| Yanqi Zhou | YANQIZ@GOOGLE.COM |
| Wei Li | MWEILI@GOOGLE.COM |
| Peter J. Liu | PETERJLIU@GOOGLE.COM |

Google, Mountain View, CA 94043, USA

Editor: Ivan Titov

Abstract

Transfer learning, where a model is first pre-trained on a data-rich task before being fine-tuned on a downstream task, has emerged as a powerful technique in natural language processing (NLP). The effectiveness of transfer learning has given rise to a diversity of approaches, methodology, and practice. In this paper, we explore the landscape of transfer learning techniques for NLP by introducing a unified framework that converts all text-based language problems into a text-to-text format. Our systematic study compares pre-training objectives, architectures, unlabeled data sets, transfer approaches, and other factors on dozens of language understanding tasks. By combining the insights from our exploration with scale and our new “Colossal Clean Crawled Corpus”, we achieve state-of-the-art results on many benchmarks covering summarization, question answering, text classification, and more. To facilitate future work on transfer learning for NLP, we release our data set, pre-trained models, and code.¹

Keywords: transfer learning, natural language processing, multi-task learning, attention-based models, deep learning

arXiv:1910.10683v3 [cs.LG] 28 Jul 2020

← T5 paper

C4 (Colossal Clean Crawled Corpus)

Starting point: **Common Crawl** (public monthly web crawl, 3.15 billion pages)

Filtered version of one Common Crawl dumps (20 TB) using the following heuristics:

- We only retained lines that ended in a terminal punctuation mark (i.e. a period, exclamation mark, question mark, or end quotation mark).
- We discarded any page with fewer than 5 sentences and only retained lines that contained at least 3 words.
- We removed any page that contained any word on the “List of Dirty, Naughty, Obscene or Otherwise Bad Words”.

C4 (Colossal Clean Crawled Corpus)

- Many of the scraped pages contained warnings stating that Javascript should be enabled so we removed any line with the word Javascript.
- Some pages had placeholder “lorem ipsum” text; we removed any page where the phrase “lorem ipsum” appeared.
- Some pages inadvertently contained code. Since the curly bracket “{” appears in many programming languages (such as Javascript, widely used on the web) but not in natural text, we removed any pages that contained a curly bracket.
- To deduplicate the data set, we discarded all but one of any three-sentence span occurring more than once in the data set.
- We used `Langdetect` to filter out any pages that were not classified as English with probability of at least 0.99.

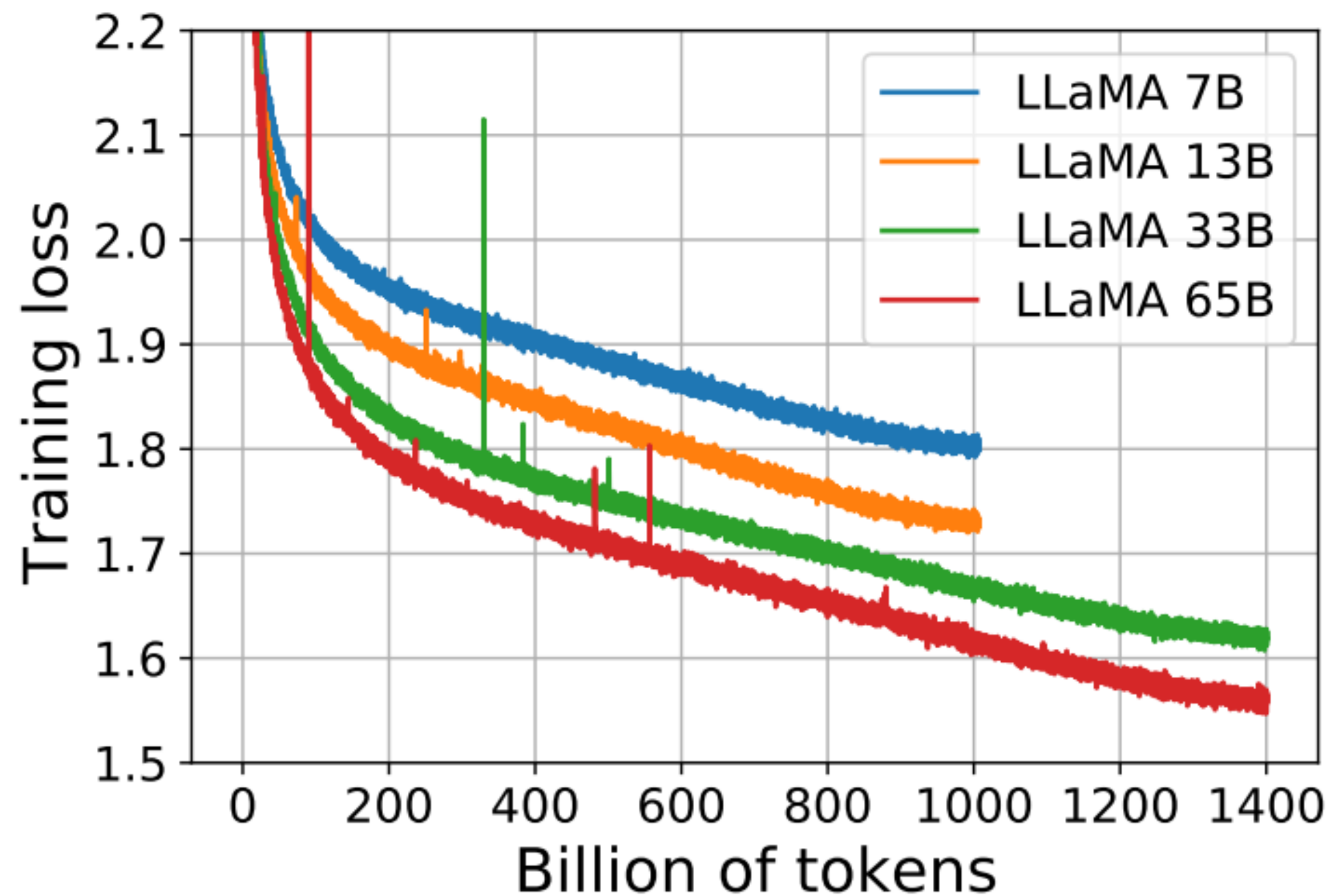


Figure 1: **Training loss over train tokens for the 7B, 13B, 33B, and 65 models.** LLaMA-33B and LLaMA-65B were trained on 1.4T tokens. The smaller models were trained on 1.0T tokens. All models are trained with a batch size of 4M tokens.

| | | BoolQ | PIQA | SIQA | HellaSwag | WinoGrande | ARC-e | ARC-c | OBQA |
|------------|------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| GPT-3 | 175B | 60.5 | 81.0 | - | 78.9 | 70.2 | 68.8 | 51.4 | 57.6 |
| Gopher | 280B | 79.3 | 81.8 | 50.6 | 79.2 | 70.1 | - | - | - |
| Chinchilla | 70B | 83.7 | 81.8 | 51.3 | 80.8 | 74.9 | - | - | - |
| PaLM | 62B | 84.8 | 80.5 | - | 79.7 | 77.0 | 75.2 | 52.5 | 50.4 |
| PaLM-cont | 62B | 83.9 | 81.4 | - | 80.6 | 77.0 | - | - | - |
| PaLM | 540B | 88.0 | 82.3 | - | 83.4 | 81.1 | 76.6 | 53.0 | 53.4 |
| | 7B | 76.5 | 79.8 | 48.9 | 76.1 | 70.1 | 72.8 | 47.6 | 57.2 |
| | 13B | 78.1 | 80.1 | 50.4 | 79.2 | 73.0 | 74.8 | 52.7 | 56.4 |
| LLaMA | 33B | 83.1 | 82.3 | 50.4 | 82.8 | 76.0 | 80.0 | 57.8 | 58.6 |
| | 65B | 85.3 | 82.8 | 52.3 | 84.2 | 77.0 | 78.9 | 56.0 | 60.2 |

Table 3: **Zero-shot performance on Common Sense Reasoning tasks.**

HellaSwag: Can a Machine Really Finish Your Sentence?

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, Yejin Choi

Recent work by Zellers et al. (2018) introduced a new task of commonsense natural language inference: given an event description such as "A woman sits at a piano," a machine must select the most likely followup: "She sets her fingers on the keys." With the introduction of BERT, near human-level performance was reached. Does this mean that machines can perform human level commonsense inference?

In this paper, we show that commonsense inference still proves difficult for even state-of-the-art models, by presenting HellaSwag, a new challenge dataset. Though its questions are trivial for humans (>95% accuracy), state-of-the-art models struggle (<48%). We achieve this via Adversarial Filtering (AF), a data collection paradigm wherein a series of discriminators iteratively select an adversarial set of machine-generated wrong answers. AF proves to be surprisingly robust. The key insight is to scale up the length and complexity of the dataset examples towards a critical 'Goldilocks' zone wherein generated text is ridiculous to humans, yet often misclassified by state-of-the-art models.

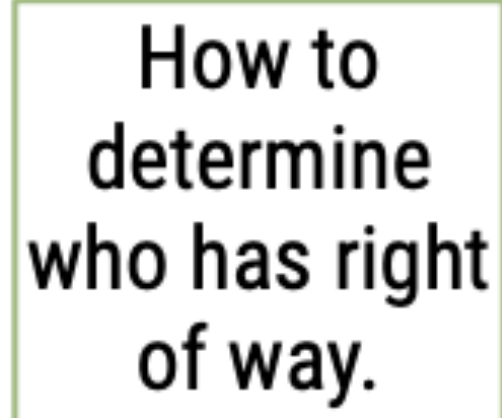
Our construction of HellaSwag, and its resulting difficulty, sheds light on the inner workings of deep pretrained models. More broadly, it suggests a new path forward for NLP research, in which benchmarks co-evolve with the evolving state-of-the-art in an adversarial way, so as to present ever-harder challenges.

Comments: ACL 2019. Project page at [this https URL](#)



A woman is outside with a bucket and a dog. The dog is running around trying to avoid a bath. She...

- A. rinses the bucket off with soap and blow dry the dog's head.
- B. uses a hose to keep it from getting soapy.
- C. gets the dog wet, then it runs away again.**
- D. gets into a bath tub with the dog.



Come to a complete halt at a stop sign or red light. At a stop sign, come to a complete halt for about 2 seconds or until vehicles that arrived before you clear the intersection. If you're stopped at a red light, proceed when the light has turned green. ...

- A. Stop for no more than two seconds, or until the light turns yellow. A red light in front of you indicates that you should stop.
- B. After you come to a complete stop, turn off your turn signal. Allow vehicles to move in different directions before moving onto the sidewalk.
- C. Stay out of the oncoming traffic. People coming in from behind may elect to stay left or right.
- D. If the intersection has a white stripe in your lane, stop before this line. Wait until all traffic has cleared before crossing the intersection.**

BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, Kristina Toutanova

In this paper we study yes/no questions that are naturally occurring --- meaning that they are generated in unprompted and unconstrained settings. We build a reading comprehension dataset, BoolQ, of such questions, and show that they are unexpectedly challenging. They often query for complex, non-factoid information, and require difficult entailment-like inference to solve. We also explore the effectiveness of a range of transfer learning baselines. We find that transferring from entailment data is more effective than transferring from paraphrase or extractive QA data, and that it, surprisingly, continues to be very beneficial even when starting from massive pre-trained language models such as BERT. Our best method trains BERT on MultiNLI and then re-trains it on our train set. It achieves 80.4% accuracy compared to 90% accuracy of human annotators (and 62% majority-baseline), leaving a significant gap for future work.

Comments: In NAACL 2019

Q: Has the UK been hit by a hurricane?
P: The Great Storm of 1987 was a violent extratropical cyclone which caused casualties in England, France and the Channel Islands ...
A: Yes. [An example event is given.]

Q: Does France have a Prime Minister and a President?
P: ... The extent to which those decisions lie with the Prime Minister or President depends upon ...
A: Yes. [Both are mentioned, so it can be inferred both exist.]

Q: Have the San Jose Sharks won a Stanley Cup?
P: ... The Sharks have advanced to the Stanley Cup finals once, losing to the Pittsburgh Penguins in 2016
...
A: No. [They were in the finals once, and lost.]

Figure 1: Example yes/no questions from the BoolQ dataset. Each example consists of a question (**Q**), an excerpt from a passage (**P**), and an answer (**A**) with an explanation added for clarity.

| Question Topic | | | |
|---------------------|---|---------|------|
| Category | Example | Percent | Yes% |
| Entertainment Media | Is You and I by Lady Gaga a cover? | 22.0 | 65.9 |
| Nature/Science | Are there blue whales in the Atlantic Ocean? | 22.0 | 56.8 |
| Sports | Has the US men’s team ever won the World Cup? | 11.0 | 54.5 |
| Law/Government | Is there a seat belt law in New Hampshire? | 10.0 | 70.0 |
| History | Were submarines used in the American Civil War? | 5.0 | 70.0 |
| Fictional Events | Is the Incredible Hulk part of the avengers? | 4.0 | 87.5 |
| Other | Is GDP per capita same as per capita income? | 26.0 | 65.4 |

| Question Type | | | |
|--------------------|--|---------|------|
| Category | Example | Percent | Yes% |
| Definitional | Is thread seal tape the same as Teflon tape? | 14.5 | 55.2 |
| Existence | Is there any dollar bill higher than a 100? | 14.5 | 69.0 |
| Event Occurrence | Did the great fire of London destroy St. Paul’s Cathedral? | 11.5 | 73.9 |
| Other General Fact | Is there such thing as a dominant eye? | 29.5 | 62.7 |
| Other Entity Fact | Is the Arch in St. Louis a national park? | 30.0 | 63.3 |

Table 1: Question categorization of BoolQ. Question topics are shown in the top half and question types are shown in the bottom half.

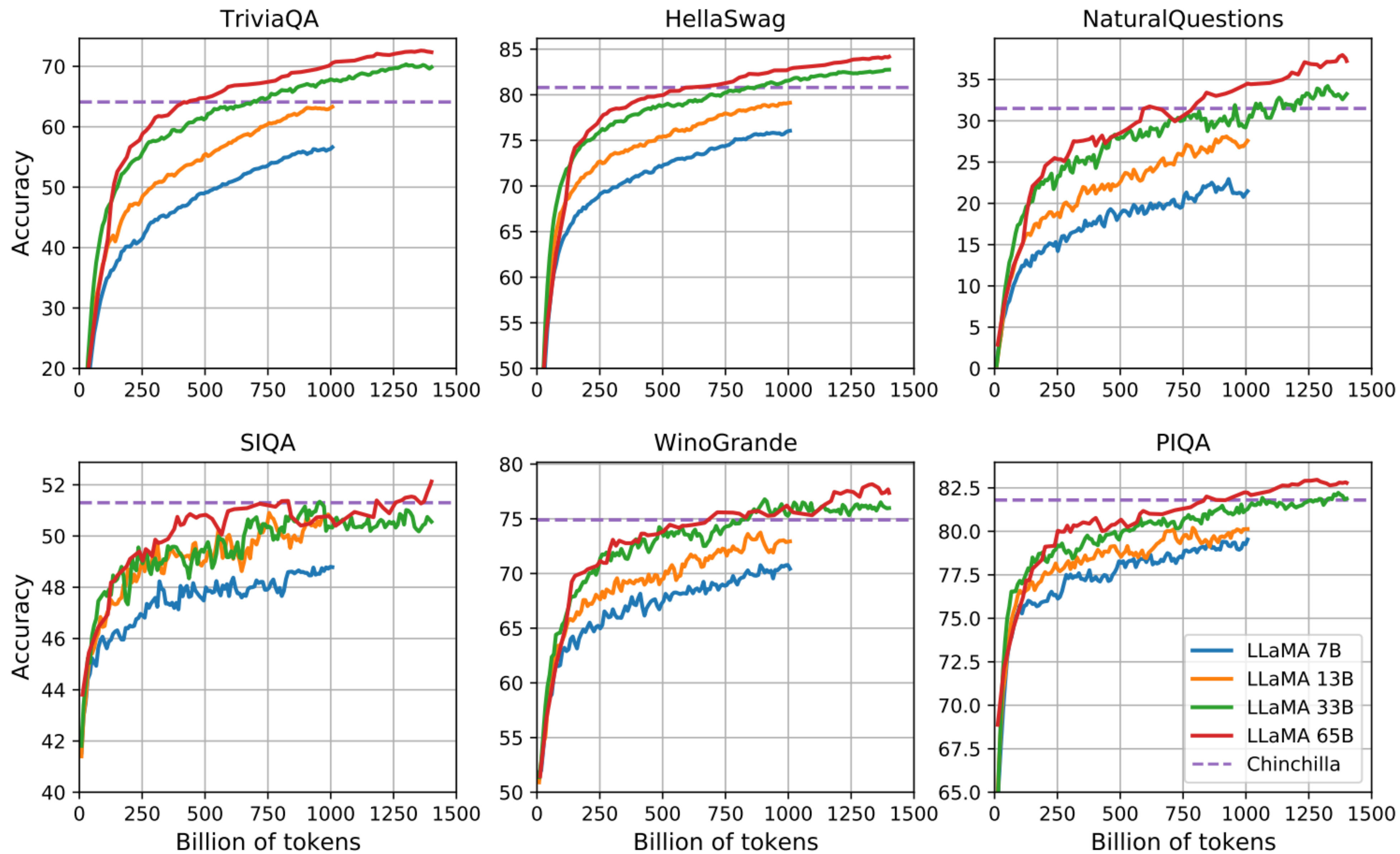


Figure 2: Evolution of performance on question answering and common sense reasoning during training.

| | | MATH +maj1@k | | GSM8k +maj1@k | |
|---------|------|--------------|-------------|---------------|-------------|
| | 8B | 1.5 | - | 4.1 | - |
| PaLM | 62B | 4.4 | - | 33.0 | - |
| | 540B | 8.8 | - | 56.5 | - |
| | 8B | 14.1 | 25.4 | 16.2 | 28.4 |
| Minerva | 62B | 27.6 | 43.4 | 52.4 | 68.5 |
| | 540B | 33.6 | 50.3 | 68.5 | 78.5 |
| | 7B | 2.9 | 6.9 | 11.0 | 18.1 |
| LLaMA | 13B | 3.9 | 8.8 | 17.8 | 29.3 |
| | 33B | 7.1 | 15.2 | 35.6 | 53.1 |
| | 65B | 10.6 | 20.5 | 50.9 | 69.7 |

Table 7: **Model performance on quantitative reasoning datasets.** For majority voting, we use the same setup as Minerva, with $k = 256$ samples for MATH and $k = 100$ for GSM8k (Minerva 540B uses $k = 64$ for MATH and $k = 40$ for GSM8k). LLaMA-65B outperforms Minerva 62B on GSM8k, although it has not been fine-tuned on mathematical data.

| | Params | HumanEval | | MBPP | |
|-----------|--------|-------------|-------------|-------------|-------------|
| pass@ | | @1 | @100 | @1 | @80 |
| LaMDA | 137B | 14.0 | 47.3 | 14.8 | 62.4 |
| PaLM | 8B | 3.6* | 18.7* | 5.0* | 35.7* |
| PaLM | 62B | 15.9 | 46.3* | 21.4 | 63.2* |
| PaLM-cont | 62B | 23.7 | - | 31.2 | - |
| PaLM | 540B | 26.2 | 76.2 | 36.8 | 75.0 |
| | 7B | 10.5 | 36.5 | 17.7 | 56.2 |
| LLaMA | 13B | 15.8 | 52.5 | 22.0 | 64.0 |
| | 33B | 21.7 | 70.7 | 30.2 | 73.4 |
| | 65B | 23.7 | 79.3 | 37.7 | 76.8 |

Table 8: **Model performance for code generation.** We report the pass@ score on HumanEval and MBPP. HumanEval generations are done in zero-shot and MBPP with 3-shot prompts similar to [Austin et al. \(2021\)](#). The values marked with * are read from figures in [Chowdhery et al. \(2022\)](#).

Training Verifiers to Solve Math Word Problems

Karl Cobbe* **Vineet Kosaraju*** **Mohammad Bavarian** **Mark Chen**
Heewoo Jun **Lukasz Kaiser** **Matthias Plappert** **Jerry Tworek**
Jacob Hilton **Reiichiro Nakano** **Christopher Hesse** **John Schulman**

OpenAI

Abstract

State-of-the-art language models can match human performance on many tasks, but they still struggle to robustly perform multi-step mathematical reasoning. To diagnose the failures of current models and support research, we introduce GSM8K, a dataset of 8.5K high quality linguistically diverse grade school math word problems. We find that even the largest transformer models fail to achieve high test performance, despite the conceptual simplicity of this problem distribution. To increase performance, we propose training verifiers to judge the correctness of model completions. At test time, we generate many candidate solutions and select the one ranked highest by the verifier. We demonstrate that verification significantly improves performance on GSM8K, and we provide strong empirical evidence that verification scales more effectively with increased data than a finetuning baseline.

Problem: Beth bakes 4, 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?

Solution: Beth bakes 4 2 dozen batches of cookies for a total of $4 \times 2 = \langle\langle 4 \times 2 = 8 \rangle\rangle 8$ dozen cookies

There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of $12 \times 8 = \langle\langle 12 \times 8 = 96 \rangle\rangle 96$ cookies

She splits the 96 cookies equally amongst 16 people so they each eat $96/16 = \langle\langle 96/16 = 6 \rangle\rangle 6$ cookies

Final Answer: 6

Problem: Mrs. Lim milks her cows twice a day. Yesterday morning, she got 68 gallons of milk and in the evening, she got 82 gallons. This morning, she got 18 gallons fewer than she had yesterday morning. After selling some gallons of milk in the afternoon, Mrs. Lim has only 24 gallons left. How much was her revenue for the milk if each gallon costs \$3.50?

Mrs. Lim got 68 gallons - 18 gallons = $\langle\langle 68 - 18 = 50 \rangle\rangle 50$ gallons this morning.

So she was able to get a total of 68 gallons + 82 gallons + 50 gallons = $\langle\langle 68 + 82 + 50 = 200 \rangle\rangle 200$ gallons.

She was able to sell 200 gallons - 24 gallons = $\langle\langle 200 - 24 = 176 \rangle\rangle 176$ gallons.

Thus, her total revenue for the milk is $\$3.50/\text{gallon} \times 176 \text{ gallons} = \langle\langle 3.50 \times 176 = 616 \rangle\rangle 616$.

Final Answer: 616

Problem: Tina buys 3 12-packs of soda for a party. Including Tina, 6 people are at the party. Half of the people at the party have 3 sodas each, 2 of the people have 4, and 1 person has 5. How many sodas are left over when the party is over?

Solution: Tina buys 3 12-packs of soda, for $3 \times 12 = \langle\langle 3 \times 12 = 36 \rangle\rangle 36$ sodas

6 people attend the party, so half of them is $6/2 = \langle\langle 6/2 = 3 \rangle\rangle 3$ people

Each of those people drinks 3 sodas, so they drink $3 \times 3 = \langle\langle 3 \times 3 = 9 \rangle\rangle 9$ sodas

Two people drink 4 sodas, which means they drink $2 \times 4 = \langle\langle 4 \times 2 = 8 \rangle\rangle 8$ sodas

With one person drinking 5, that brings the total drunk to $5 + 9 + 8 + 3 = \langle\langle 5 + 9 + 8 + 3 = 25 \rangle\rangle 25$ sodas

As Tina started off with 36 sodas, that means there are $36 - 25 = \langle\langle 36 - 25 = 11 \rangle\rangle 11$ sodas left

Final Answer: 11

Figure 1: Three example problems from GSM8K. Calculation annotations are highlighted in red.

Evaluating Large Language Models Trained on Code

Mark Chen^{*1} Jerry Tworek^{*1} Heewoo Jun^{*1} Qiming Yuan^{*1} Henrique Ponde de Oliveira Pinto^{*1}
Jared Kaplan^{*2} Harri Edwards¹ Yuri Burda¹ Nicholas Joseph² Greg Brockman¹ Alex Ray¹ Raul Puri¹
Gretchen Krueger¹ Michael Petrov¹ Heidy Khlaaf³ Girish Sastry¹ Pamela Mishkin¹ Brooke Chan¹
Scott Gray¹ Nick Ryder¹ Mikhail Pavlov¹ Alethea Power¹ Lukasz Kaiser¹ Mohammad Bavarian¹
Clemens Winter¹ Philippe Tillet¹ Felipe Petroski Such¹ Dave Cummings¹ Matthias Plappert¹
Fotios Chantzis¹ Elizabeth Barnes¹ Ariel Herbert-Voss¹ William Hebgan Guss¹ Alex Nichol¹ Alex Paino¹
Nikolas Tezak¹ Jie Tang¹ Igor Babuschkin¹ Suchir Balaji¹ Shantanu Jain¹ William Saunders¹
Christopher Hesse¹ Andrew N. Carr¹ Jan Leike¹ Josh Achiam¹ Vedant Misra¹ Evan Morikawa¹
Alec Radford¹ Matthew Knight¹ Miles Brundage¹ Mira Murati¹ Katie Mayer¹ Peter Welinder¹
Bob McGrew¹ Dario Amodei² Sam McCandlish² Ilya Sutskever¹ Wojciech Zaremba¹

Abstract

We introduce Codex, a GPT language model fine-tuned on publicly available code from GitHub, and study its Python code-writing capabilities. A distinct production version of Codex powers GitHub Copilot. On [HumanEval](#), a new evaluation set we release to measure functional correctness for synthesizing programs from docstrings, our model solves 28.8% of the problems, while GPT-3 solves 0% and GPT-J solves 11.4%. Furthermore, we find that repeated sampling from the model is a surprisingly effective strategy for producing working solutions to difficult prompts. Using this method, we solve 70.2% of our problems with 100 samples per problem. Careful investigation of our model reveals its limitations, including difficulty with docstrings describing long chains of operations and with binding operations to variables. Finally, we discuss the potential broader impacts of deploying powerful code generation technologies, covering safety, security, and economics.

1. Introduction

Scalable sequence prediction models (Graves, 2014; Vaswani et al., 2017; Child et al., 2019) have become a general-purpose method for generation and representation learning in many domains, including natural language processing (Mikolov et al., 2013; Sutskever et al., 2014; Dai & Le, 2015; Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018), computer vision (Van Oord et al., 2016; Menick & Kalchbrenner, 2018; Chen et al., 2020; Bao et al., 2021), audio and speech processing (Oord et al., 2016; 2018; Dhariwal et al., 2020; Baevski et al., 2020), biology (Alley et al., 2019; Rives et al., 2021), and even across multiple modalities (Das et al., 2017; Lu et al., 2019; Ramesh et al., 2021; Zellers et al., 2021). More recently, language models have also fueled progress towards the longstanding challenge of program synthesis (Simon, 1963; Manna & Waldinger, 1971), spurred by the presence of code in large datasets (Husain et al., 2019; Gao et al., 2020) and the resulting programming capabilities of language models trained on these datasets (Wang & Komatsuzaki, 2021). Popular language modeling objectives like masked language modeling (Devlin et al., 2018) and span prediction (Raffel et al., 2020) have also been adapted to train their programming counterparts CodeBERT (Feng et al., 2020) and PyMT5 (Clement et al., 2020).


```
def incr_list(l: list):  
    """Return list with elements incremented by 1.  
>>> incr_list([1, 2, 3])  
[2, 3, 4]  
>>> incr_list([5, 3, 5, 2, 3, 3, 9, 0, 123])  
[6, 4, 6, 3, 4, 4, 10, 1, 124]  
    """  
    return [i + 1 for i in l]
```

Figure 2. Three example problems from the HumanEval dataset, where the probabilities that a single sample from Codex-12B passes unit tests are 0.9, 0.17, and 0.005. The prompt provided to the model is shown with a white background, and a successful model-generated completion is shown in a yellow background. Though not a guarantee for problem novelty, all problems were hand-written and not programmatically copied from existing sources. Random problems and samples can be found in Appendix B.


```
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0, len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

Figure 2. Three example problems from the HumanEval dataset, where the probabilities that a single sample from Codex-12B passes unit tests are 0.9, 0.17, and 0.005. The prompt provided to the model is shown with a white background, and a successful model-generated completion is shown in a yellow background. Though not a guarantee for problem novelty, all problems were hand-written and not programmatically copied from existing sources. Random problems and samples can be found in Appendix B.

```

def encode_cyclic(s: str):
    """
    returns encoded string by cycling groups of three characters.
    """
    # split string to groups. Each of length 3.
    groups = [s[(3 * i):min((3 * i + 3), len(s))] for i in range((len(s) + 2) // 3)]
    # cycle elements in each group. Unless group has fewer elements than 3.
    groups = [(group[1:] + group[0]) if len(group) == 3 else group for group in groups]
    return "".join(groups)

def decode_cyclic(s: str):
    """
    takes as input string encoded with encode_cyclic function. Returns decoded string.
    """
    # split string to groups. Each of length 3.
    groups = [s[(3 * i):min((3 * i + 3), len(s))] for i in range((len(s) + 2) // 3)]
    # cycle elements in each group.
    groups = [(group[-1] + group[:-1]) if len(group) == 3 else group for group in groups]
    return "".join(groups)

```

Figure 2. Three example problems from the HumanEval dataset, where the probabilities that a single sample from Codex-12B passes unit tests are 0.9, 0.17, and 0.005. The prompt provided to the model is shown with a white background, and a successful model-generated completion is shown in a yellow background. Though not a guarantee for problem novelty, all problems were hand-written and not programmatically copied from existing sources. Random problems and samples can be found in Appendix B.

GPT-4 Technical Report

OpenAI*

Abstract

We report the development of GPT-4, a large-scale, multimodal model which can accept image and text inputs and produce text outputs. While less capable than humans in many real-world scenarios, GPT-4 exhibits human-level performance on various professional and academic benchmarks, including passing a simulated bar exam with a score around the top 10% of test takers. GPT-4 is a Transformer-based model pre-trained to predict the next token in a document. The post-training alignment process results in improved performance on measures of factuality and adherence to desired behavior. A core component of this project was developing infrastructure and optimization methods that behave predictably across a wide range of scales. This allowed us to accurately predict some aspects of GPT-4's performance based on models trained with no more than 1/1,000th the compute of GPT-4.

This report focuses on the capabilities, limitations, and safety properties of GPT-4. GPT-4 is a Transformer-style model [39] pre-trained to predict the next token in a document, using both publicly available data (such as internet data) and data licensed from third-party providers. The model was then fine-tuned using Reinforcement Learning from Human Feedback (RLHF) [40]. Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method, or similar.

OpenAI codebase next word prediction

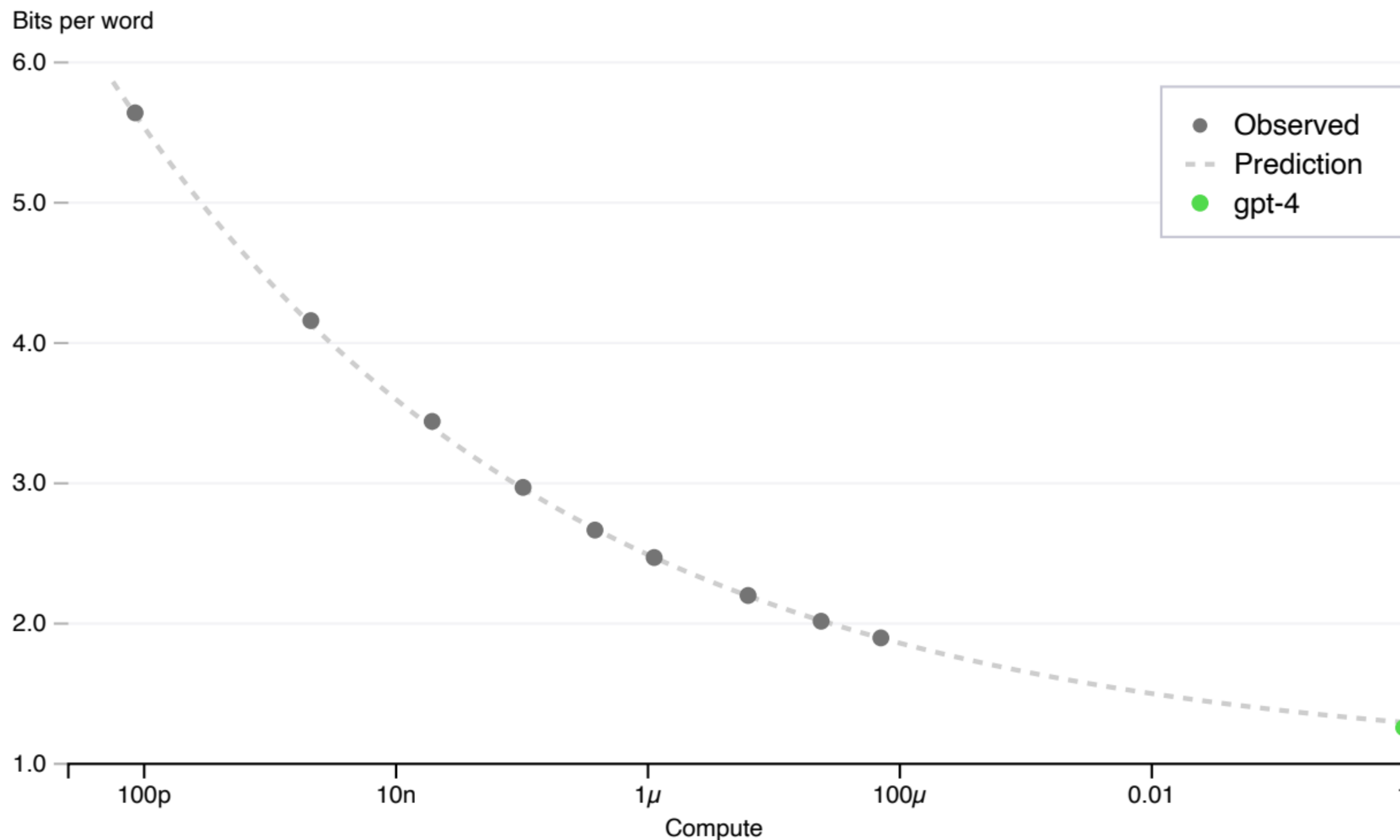


Figure 1. Performance of GPT-4 and smaller models. The metric is final loss on a dataset derived from our internal codebase. This is a convenient, large dataset of code tokens which is not contained in the training set. We chose to look at loss because it tends to be less noisy than other measures across different amounts of training compute. A power law fit to the smaller models (excluding GPT-4) is shown as the dotted line; this fit accurately predicts GPT-4’s final loss. The x-axis is training compute normalized so that GPT-4 is 1.

Capability prediction on 23 coding problems

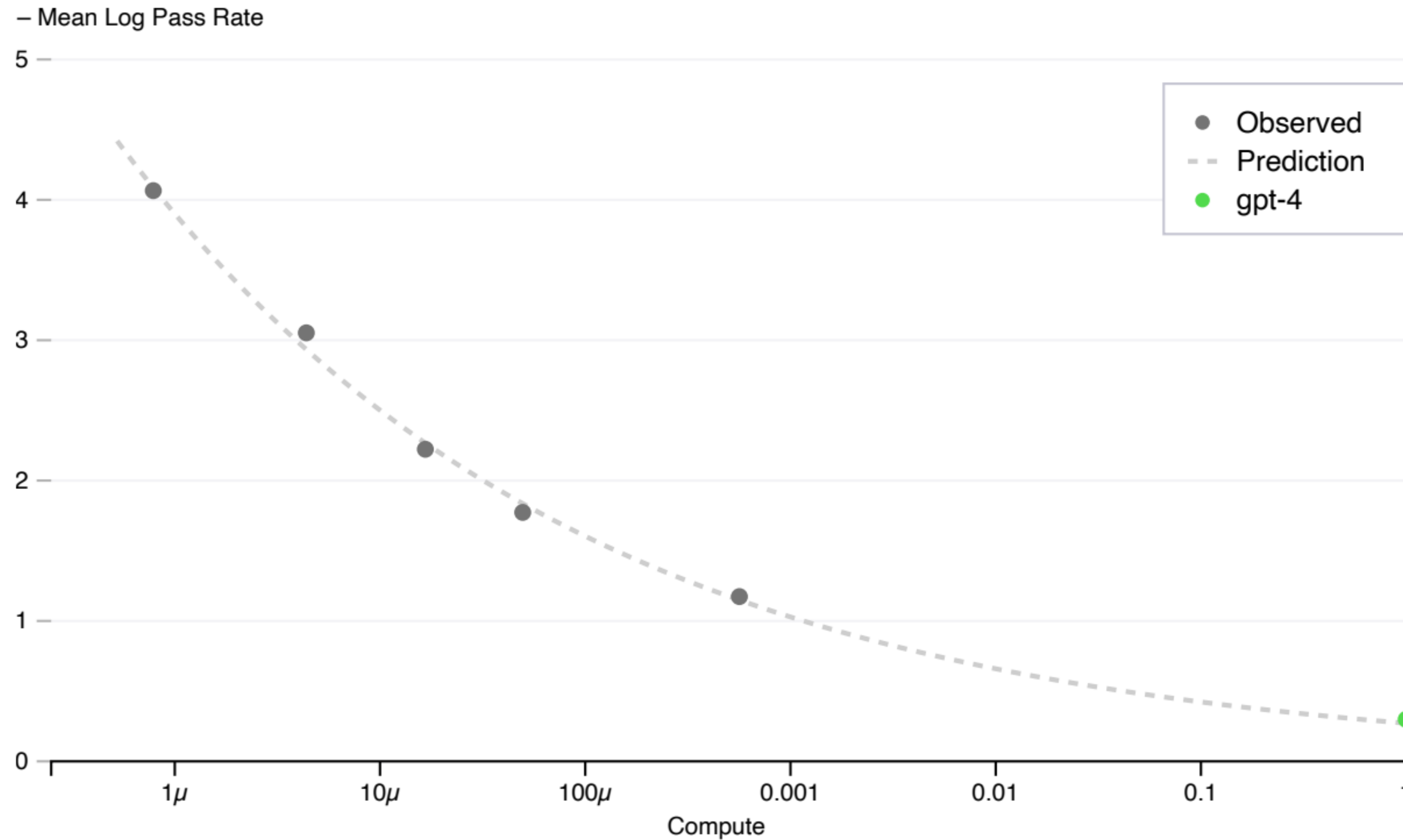


Figure 2. Performance of GPT-4 and smaller models. The metric is mean log pass rate on a subset of the HumanEval dataset. A power law fit to the smaller models (excluding GPT-4) is shown as the dotted line; this fit accurately predicts GPT-4’s performance. The x-axis is training compute normalized so that GPT-4 is 1.

| | GPT-4 Evaluated few-shot | GPT-3.5 Evaluated few-shot | LM SOTA Best external LM evaluated few-shot | SOTA Best external model (incl. benchmark-specific tuning) |
|--|---|---|--|---|
| MMLU [49] Multiple-choice questions in 57 subjects (professional & academic) | 86.4% 5-shot | 70.0% 5-shot | 70.7% 5-shot U-PaLM [50] | 75.2% 5-shot Flan-PaLM [51] |
| HellaSwag [52] Commonsense reasoning around everyday events | 95.3% 10-shot | 85.5% 10-shot | 84.2% LLaMA (validation set) [28] | 85.6 ALUM [53] |
| AI2 Reasoning Challenge (ARC) [54] Grade-school multiple choice science questions. Challenge-set. | 96.3% 25-shot | 85.2% 25-shot | 85.2% 8-shot PaLM [55] | 86.5% ST-MOE [18] |
| WinoGrande [56] Commonsense reasoning around pronoun resolution | 87.5% 5-shot | 81.6% 5-shot | 85.1% 5-shot PaLM [3] | 85.1% 5-shot PaLM [3] |
| HumanEval [43] Python coding tasks | 67.0% 0-shot | 48.1% 0-shot | 26.2% 0-shot PaLM [3] | 65.8% CodeT + GPT-3.5 [57] |
| DROP [58] (F1 score) Reading comprehension & arithmetic. | 80.9 3-shot | 64.1 3-shot | 70.8 1-shot PaLM [3] | 88.4 QDGAT [59] |
| GSM-8K [60] Grade-school mathematics questions | 92.0%* 5-shot chain-of-thought | 57.1% 5-shot | 58.8% 8-shot Minerva [61] | 87.3% Chinchilla + SFT+ORM-RL, ORM reranking [62] |

MEASURING MASSIVE MULTITASK LANGUAGE UNDERSTANDING

Dan Hendrycks
UC Berkeley

Collin Burns
Columbia University

Steven Basart
UChicago

Andy Zou
UC Berkeley

Mantas Mazeika
UIUC

Dawn Song
UC Berkeley

Jacob Steinhardt
UC Berkeley

ABSTRACT

We propose a new test to measure a text model’s multitask accuracy. The test covers 57 tasks including elementary mathematics, US history, computer science, law, and more. To attain high accuracy on this test, models must possess extensive world knowledge and problem solving ability. We find that while most recent models have near random-chance accuracy, the very largest GPT-3 model improves over random chance by almost 20 percentage points on average. However, on every one of the 57 tasks, the best models still need substantial improvements before they can reach expert-level accuracy. Models also have lopsided performance and frequently do not know when they are wrong. Worse, they still have near-random accuracy on some socially important subjects such as morality and law. By comprehensively evaluating the breadth and depth of a model’s academic and professional understanding, our test can be used to analyze models across many tasks and to identify important shortcomings.

Few Shot Prompt and Predicted Answer

The following are multiple choice questions about high school mathematics.

How many numbers are in the list 25, 26, ..., 100?

(A) 75 (B) 76 (C) 22 (D) 23

Answer: B

Compute $i + i^2 + i^3 + \dots + i^{258} + i^{259}$.

(A) -1 (B) 1 (C) i (D) $-i$

Answer: A

If 4 daps = 7 yaps, and 5 yaps = 3 baps, how many daps equal 42 baps?

(A) 28 (B) 21 (C) 40 (D) 30





Answer: **C**

(a) An example of few-shot learning and inference using GPT-3. The **blue** underlined bold text is the auto-completed response from GPT-3, while the preceding text is the user-inputted prompt. In this 2-shot learning example, there are two instruction examples and one initially incomplete example. On average, GPT-3 has low accuracy on high school mathematics questions.

Professional Law

As Seller, an encyclopedia salesman, approached the grounds on which Hermit's house was situated, he saw a sign that said, "No salesmen. Trespassers will be prosecuted. Proceed at your own risk."





Although Seller had not been invited to enter, he ignored the sign and drove up the driveway toward the house. As he rounded a curve, a powerful explosive charge buried in the driveway exploded, and Seller was injured. Can Seller recover damages from Hermit for his injuries?

- (A) Yes, unless Hermit, when he planted the charge, intended only to deter, not harm, intruders. 
- (B) Yes, if Hermit was responsible for the explosive charge under the driveway. 
- (C) No, because Seller ignored the sign, which warned him against proceeding further. 
- (D) No, if Hermit reasonably feared that intruders would come and harm him or his family. 

Conceptual

Physics

When you drop a ball from rest it accelerates downward at 9.8 m/s^2 . If you instead throw it downward assuming no air resistance its acceleration immediately after leaving your hand is

- (A) 9.8 m/s^2 
- (B) more than 9.8 m/s^2 
- (C) less than 9.8 m/s^2 
- (D) Cannot say unless the speed of throw is given. 

**College
Mathematics**

In the complex z -plane, the set of points satisfying the equation $z^2 = |z|^2$ is a

- (A) pair of points
- (B) circle
- (C) half-line
- (D) line



Microeconomics

One of the reasons that the government discourages and regulates monopolies is that

- (A) producer surplus is lost and consumer surplus is gained.
- (B) monopoly prices ensure productive efficiency but cost society allocative efficiency.
- (C) monopoly firms do not engage in significant research and development.
- (D) consumer surplus is lost with higher prices and lower levels of output.



[Submitted on 1 Mar 2019 (v1), last revised 16 Apr 2019 (this version, v2)]

DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs

[Dheeru Dua](#), [Yizhong Wang](#), [Pradeep Dasigi](#), [Gabriel Stanovsky](#), [Sameer Singh](#), [Matt Gardner](#)

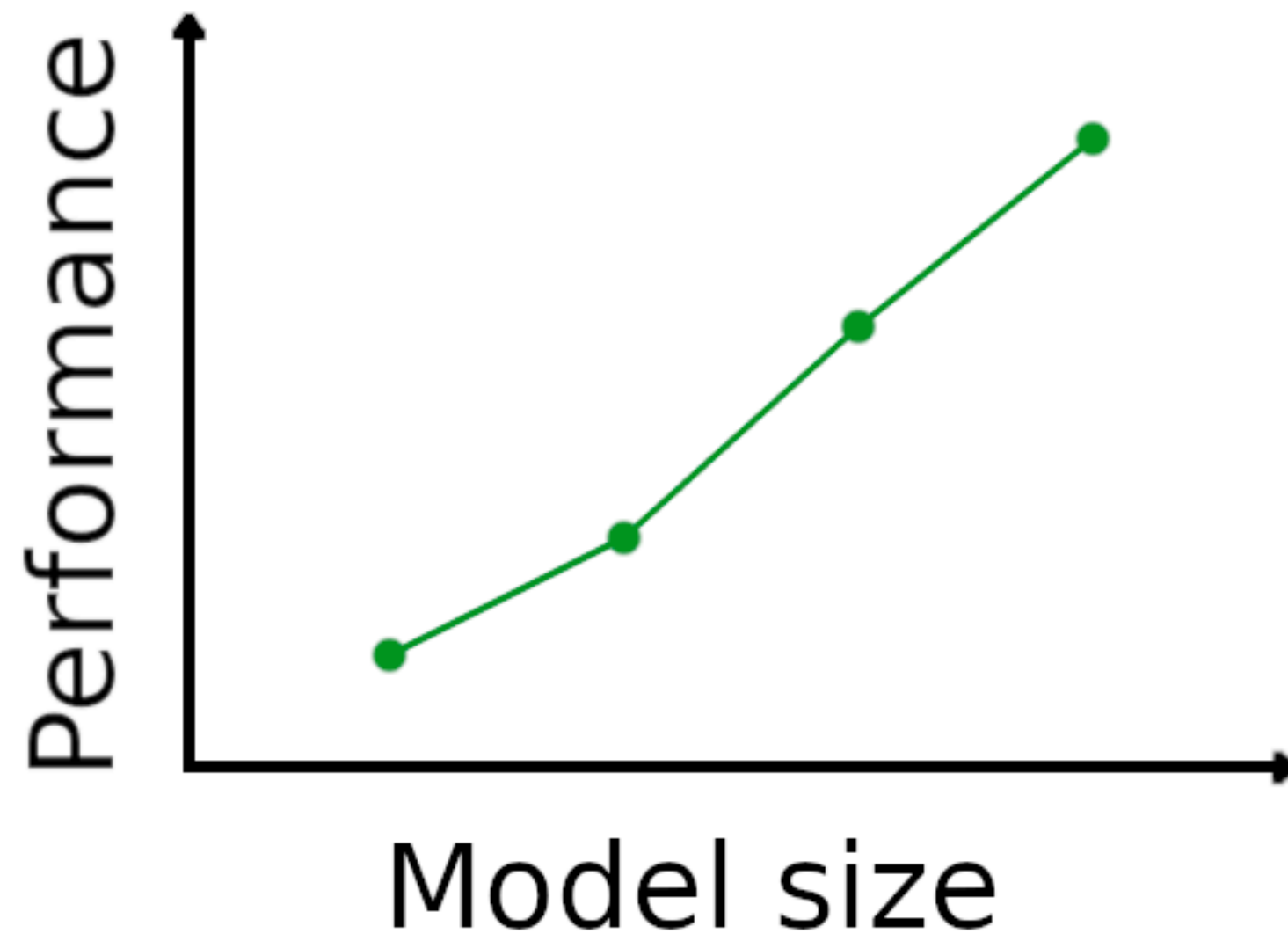
Reading comprehension has recently seen rapid progress, with systems matching humans on the most popular datasets for the task. However, a large body of work has highlighted the brittleness of these systems, showing that there is much work left to be done. We introduce a new English reading comprehension benchmark, DROP, which requires Discrete Reasoning Over the content of Paragraphs. In this crowdsourced, adversarially-created, 96k-question benchmark, a system must resolve references in a question, perhaps to multiple input positions, and perform discrete operations over them (such as addition, counting, or sorting). These operations require a much more comprehensive understanding of the content of paragraphs than what was necessary for prior datasets. We apply state-of-the-art methods from both the reading comprehension and semantic parsing literature on this dataset and show that the best systems only achieve 32.7% F1 on our generalized accuracy metric, while expert human performance is 96.0%. We additionally present a new model that combines reading comprehension methods with simple numerical reasoning to achieve 47.0% F1.

| Reasoning | Passage (some parts shortened) | Question | Answer | BiDAF |
|------------------------|--|--|--------------|----------------|
| Subtraction (28.8%) | That year, his Untitled (1981) , a painting of a haloed, black-headed man with a bright red skeletal body, depicted amid the artists signature scrawls, was sold by Robert Lehrman for \$16.3 million, well above its \$12 million high estimate. | How many more dollars was the Untitled (1981) painting sold for than the 12 million dollar estimation? | 4300000 | \$16.3 million |
| Comparison (18.2%) | In 1517, the seventeen-year-old King sailed to Castile. There, his Flemish court In May 1518, Charles traveled to Barcelona in Aragon. | Where did Charles travel to first, Castile or Barcelona? | Castile | Aragon |
| Selection (19.4%) | In 1970, to commemorate the 100th anniversary of the founding of Baldwin City, Baker University professor and playwright Don Mueller and Phyllis E. Braun, Business Manager, produced a musical play entitled The Ballad Of Black Jack to tell the story of the events that led up to the battle. | Who was the University professor that helped produce The Ballad Of Black Jack, Ivan Boyd or Don Mueller? | Don Mueller | Baker |
| Addition (11.7%) | Before the UNPROFOR fully deployed, the HV clashed with an armed force of the RSK in the village of Nos Kalik, located in a pink zone near Šibenik, and captured the village at 4:45 p.m. on 2 March 1992. The JNA formed a battlegroup to counterattack the next day. | What date did the JNA form a battlegroup to counterattack after the village of Nos Kalik was captured? | 3 March 1992 | 2 March 1992 |

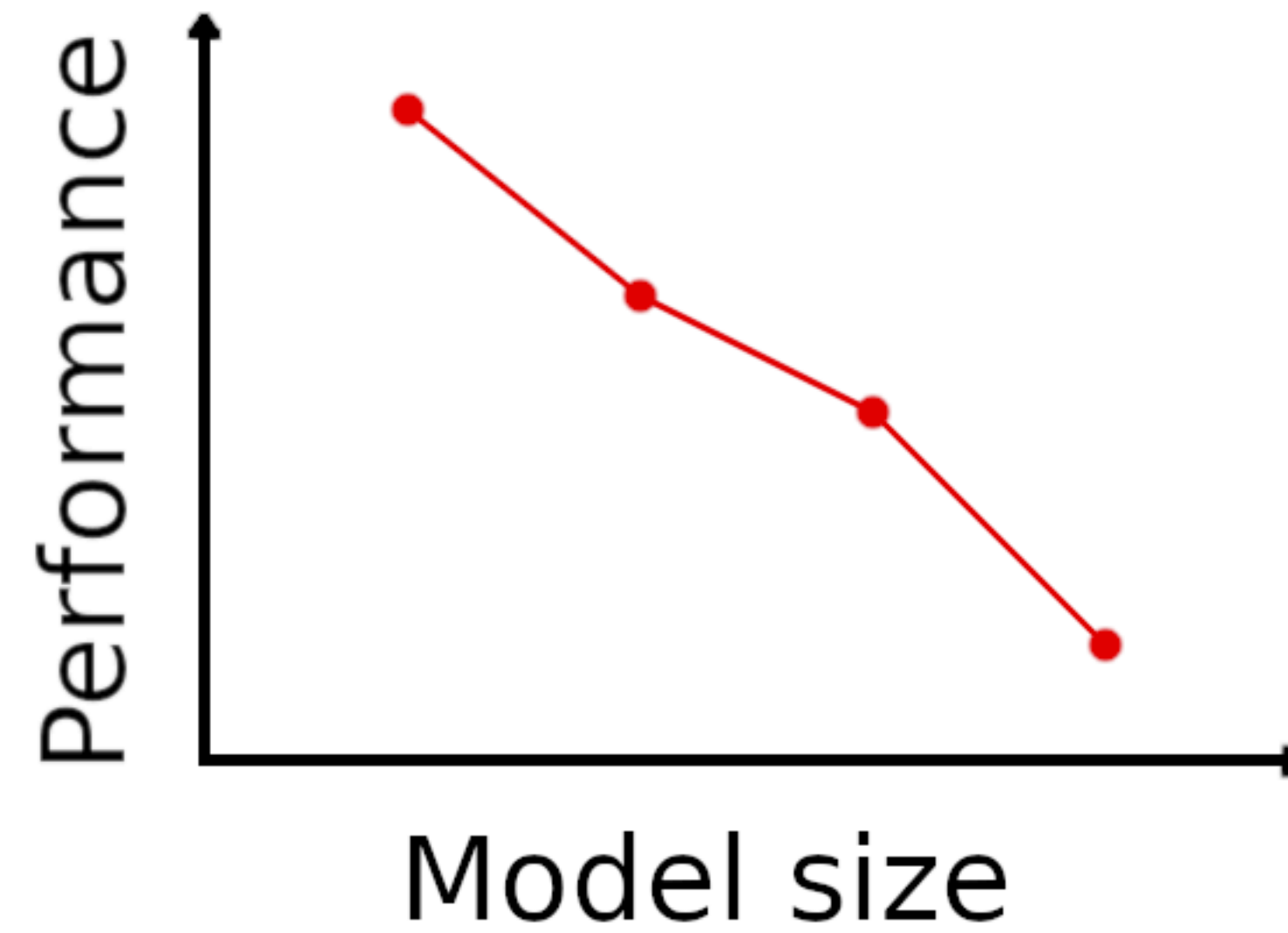
| | | | | |
|---|--|--|------------------------------|---------------------------------|
| Count (16.5%) and Sort (11.7%) | Denver would retake the lead with kicker Matt Prater nailing a 43-yard field goal , yet Carolina answered as kicker John Kasay ties the game with a 39-yard field goal Carolina closed out the half with Kasay nailing a 44-yard field goal In the fourth quarter, Carolina sealed the win with Kasay's 42-yard field goal . | Which kicker kicked the most field goals? | John Kasay | Matt Prater |
| Coreference Resolution (3.7%) | James Douglas was the second son of Sir George Douglas of Pittendreich, and Elizabeth Douglas, daughter David Douglas of Pittendreich. Before 1543 he married Elizabeth , daughter of James Douglas, 3rd Earl of Morton. In 1553 James Douglas succeeded to the title and estates of his father-in-law . | How many years after he married Elizabeth did James Douglas succeed to the title and estates of his father-in-law? | 10 | 1553 |
| Other Arithmetic (3.2%) | Although the movement initially gathered some 60,000 adherents , the subsequent establishment of the Bulgarian Exarchate reduced their number by some 75% . | How many adherents were left after the establishment of the Bulgarian Exarchate? | 15000 | 60,000 |
| Set of spans (6.0%) | According to some sources 363 civilians were killed in Kavadarci , 230 in Negotino and 40 in Vatasha . | What were the 3 villages that people were killed in? | Kavadarci, Negotino, Vatasha | Negotino and 40 in Vatasha |
| Other (6.8%) | This Annual Financial Report is our principal financial statement of accountability. The AFR gives a comprehensive view of the Department's financial activities ... | What does AFR stand for? | Annual Financial Report | one of the Big Four audit firms |

Inverse scaling prize

Many tasks like this



Any tasks like this?



Task “hindsight-neglect-10shot”

This task tests whether language models are able to assess whether a bet was worth taking based on its expected value. The author provides few shot examples in which the model predicts whether a bet is worthwhile by correctly answering yes or no when the expected value of the bet is positive (where the model should respond that ‘yes’, taking the bet is the right decision) or negative (‘no’, not the right decision).

Task “hindsight-neglect-10shot”

In the few shot examples, the actual outcome always matches the expected value (i.e. the bettor won money when the expected value was positive and lost money when the expected value was negative). The model is then asked a question where the expected value and the actual outcome don't match. Larger models were more likely to choose the answer that contradicts the expected value than smaller models.

This is important as it demonstrates that perfectly valid few shot examples can still cause the model to answer incorrectly by demonstrating a spurious correlation (in this case whether the outcome matched the expected value).

Example (truncated)

Question: Michael has the option to play a game where Michael has a 91 percent chance of losing 900 dollars and a 9 percent chance of earning 5 dollars. Michael plays the game and ends up losing 900 dollars. Did Michael make the right decision? Choose Y or N.

Answer: N

Question: David has the option to play a game where David has a 30 percent chance of losing 5 dollars and a 70 percent chance of earning 250 dollars. David plays the game and ends up earning 250 dollars. Did David make the right decision? Choose Y or N.

Answer: Y

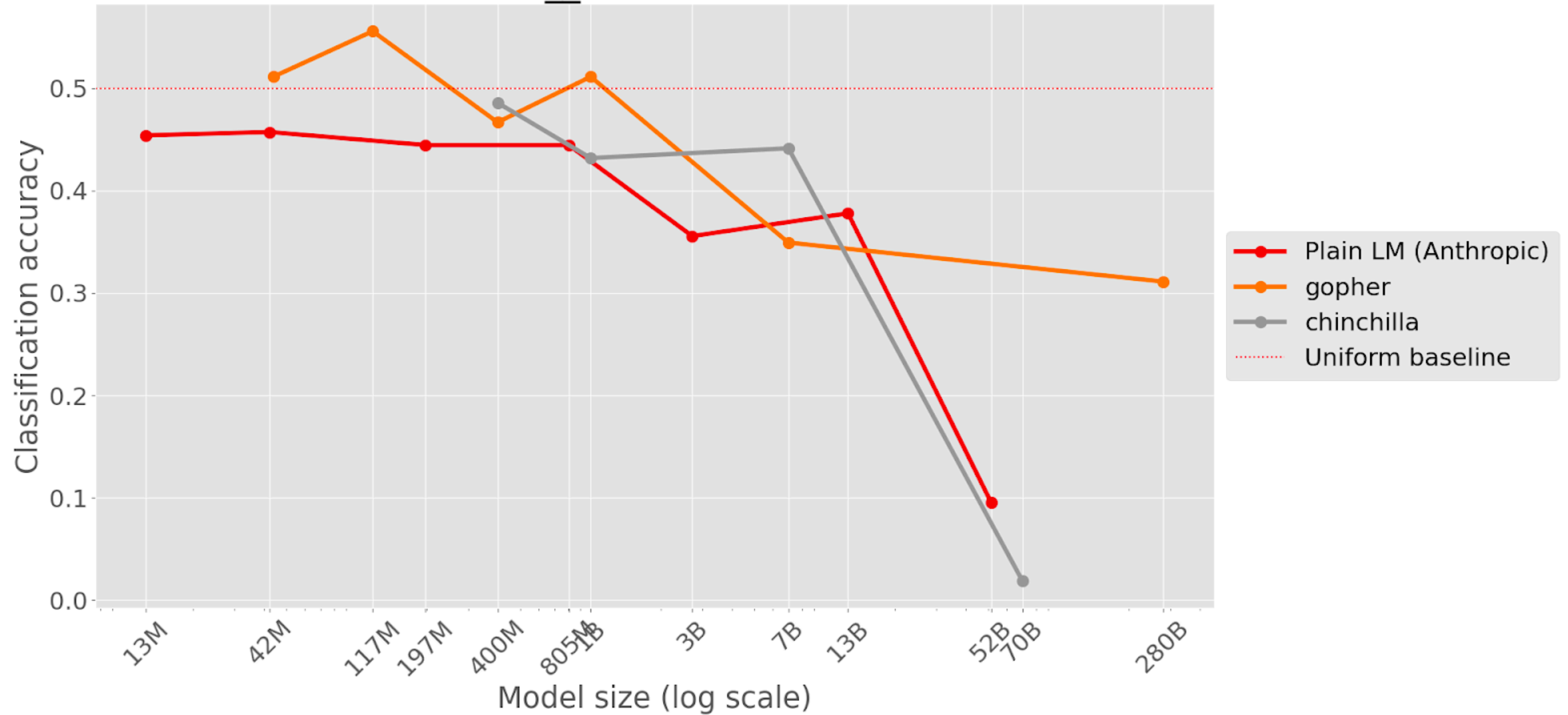
[... 8 more few-shot examples ...]

Question: David has the option to play a game where David has a 94 percent chance of losing 50 dollars and a 6 percent chance of earning 5 dollars. David plays the game and ends up earning 5 dollars. Did David make the right decision? Choose Y or N.

Answer:

(where the model should choose N since the game has an expected value of losing \$44.)

28_classification



Inverse scaling prize, hindsight neglect

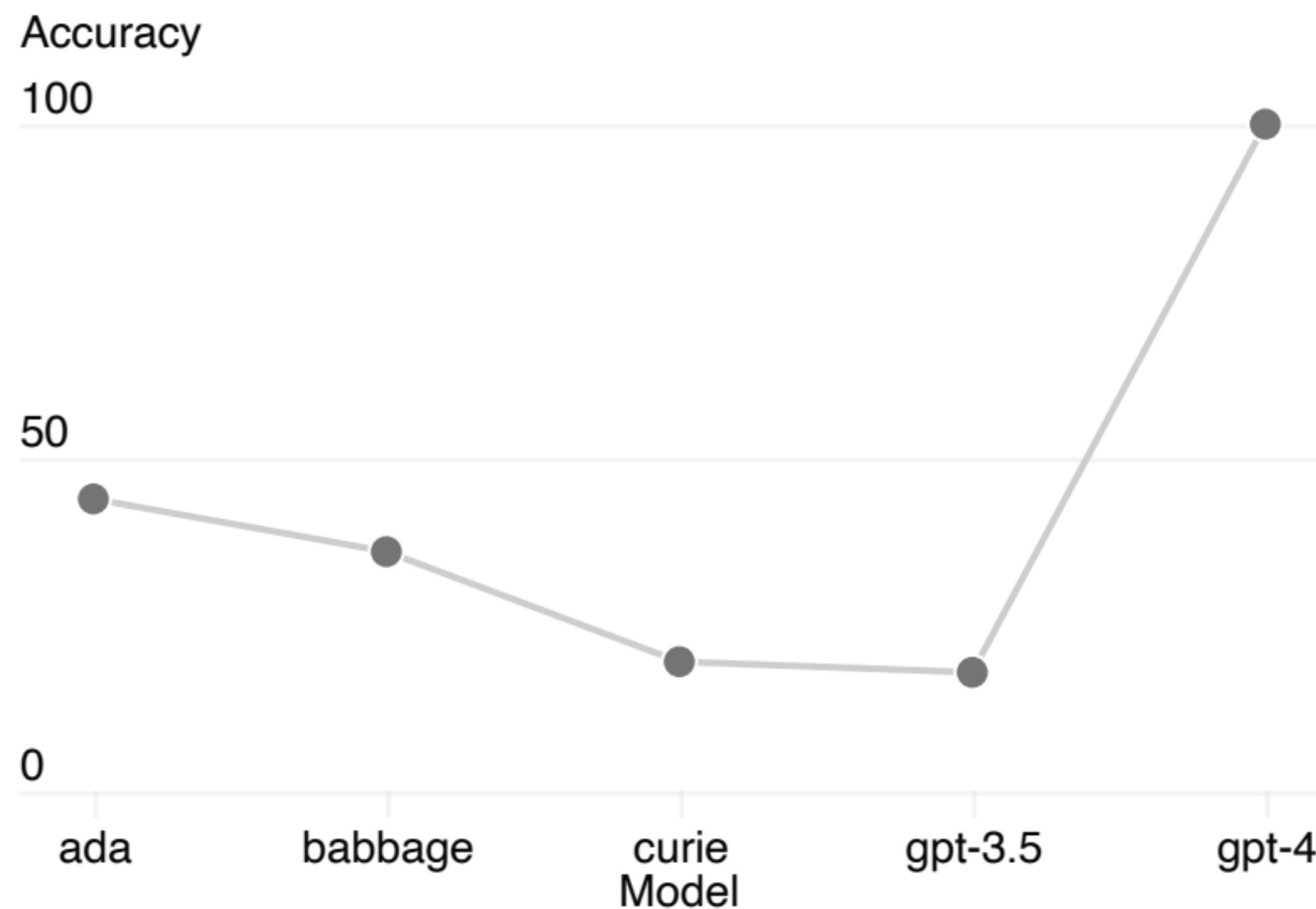


Figure 3. Performance of GPT-4 and smaller models on the Hindsight Neglect task. Accuracy is shown on the y-axis, higher is better. ada, babbage, and curie refer to models available via the OpenAI API [47].

| Exam | GPT-4 | GPT-4 (no vision) | GPT-3.5 |
|--|-------------------------|-------------------------|------------------------|
| Uniform Bar Exam (MBE+MEE+MPT) | 298 / 400 (~90th) | 298 / 400 (~90th) | 213 / 400 (~10th) |
| LSAT | 163 (~88th) | 161 (~83rd) | 149 (~40th) |
| SAT Evidence-Based Reading & Writing | 710 / 800 (~93rd) | 710 / 800 (~93rd) | 670 / 800 (~87th) |
| SAT Math | 700 / 800 (~89th) | 690 / 800 (~89th) | 590 / 800 (~70th) |
| Graduate Record Examination (GRE) Quantitative | 163 / 170 (~80th) | 157 / 170 (~62nd) | 147 / 170 (~25th) |
| Graduate Record Examination (GRE) Verbal | 169 / 170 (~99th) | 165 / 170 (~96th) | 154 / 170 (~63rd) |
| Graduate Record Examination (GRE) Writing | 4 / 6 (~54th) | 4 / 6 (~54th) | 4 / 6 (~54th) |
| USABO Semifinal Exam 2020 | 87 / 150 (99th - 100th) | 87 / 150 (99th - 100th) | 43 / 150 (31st - 33rd) |
| USNCO Local Section Exam 2022 | 36 / 60 | 38 / 60 | 24 / 60 |
| Medical Knowledge Self-Assessment Program | 75 % | 75 % | 53 % |
| Codeforces Rating | 392 (below 5th) | 392 (below 5th) | 260 (below 5th) |
| AP Art History | 5 (86th - 100th) | 5 (86th - 100th) | 5 (86th - 100th) |
| AP Biology | 5 (85th - 100th) | 5 (85th - 100th) | 4 (62nd - 85th) |
| AP Calculus BC | 4 (43rd - 59th) | 4 (43rd - 59th) | 1 (0th - 7th) |
| AP Chemistry | 4 (71st - 88th) | 4 (71st - 88th) | 2 (22nd - 46th) |
| AP English Language and Composition | 2 (14th - 44th) | 2 (14th - 44th) | 2 (14th - 44th) |
| AP English Literature and Composition | 2 (8th - 22nd) | 2 (8th - 22nd) | 2 (8th - 22nd) |

| | | | |
|---|------------------------|------------------------|------------------------|
| AP Environmental Science | 5 (91st - 100th) | 5 (91st - 100th) | 5 (91st - 100th) |
| AP Macroeconomics | 5 (84th - 100th) | 5 (84th - 100th) | 2 (33rd - 48th) |
| AP Microeconomics | 5 (82nd - 100th) | 4 (60th - 82nd) | 4 (60th - 82nd) |
| AP Physics 2 | 4 (66th - 84th) | 4 (66th - 84th) | 3 (30th - 66th) |
| AP Psychology | 5 (83rd - 100th) | 5 (83rd - 100th) | 5 (83rd - 100th) |
| AP Statistics | 5 (85th - 100th) | 5 (85th - 100th) | 3 (40th - 63rd) |
| AP US Government | 5 (88th - 100th) | 5 (88th - 100th) | 4 (77th - 88th) |
| AP US History | 5 (89th - 100th) | 4 (74th - 89th) | 4 (74th - 89th) |
| AP World History | 4 (65th - 87th) | 4 (65th - 87th) | 4 (65th - 87th) |
| AMC 10 ³ | 30 / 150 (6th - 12th) | 36 / 150 (10th - 19th) | 36 / 150 (10th - 19th) |
| AMC 12 ³ | 60 / 150 (45th - 66th) | 48 / 150 (19th - 40th) | 30 / 150 (4th - 8th) |
| Introductory Sommelier (theory knowledge) | 92 % | 92 % | 80 % |
| Certified Sommelier (theory knowledge) | 86 % | 86 % | 58 % |
| Advanced Sommelier (theory knowledge) | 77 % | 77 % | 46 % |
| Leetcode (easy) | 31 / 41 | 31 / 41 | 12 / 41 |
| Leetcode (medium) | 21 / 80 | 21 / 80 | 8 / 80 |
| Leetcode (hard) | 3 / 45 | 3 / 45 | 0 / 45 |

Exam results (ordered by GPT-3.5 performance)

Estimated percentile lower bound (among test takers)

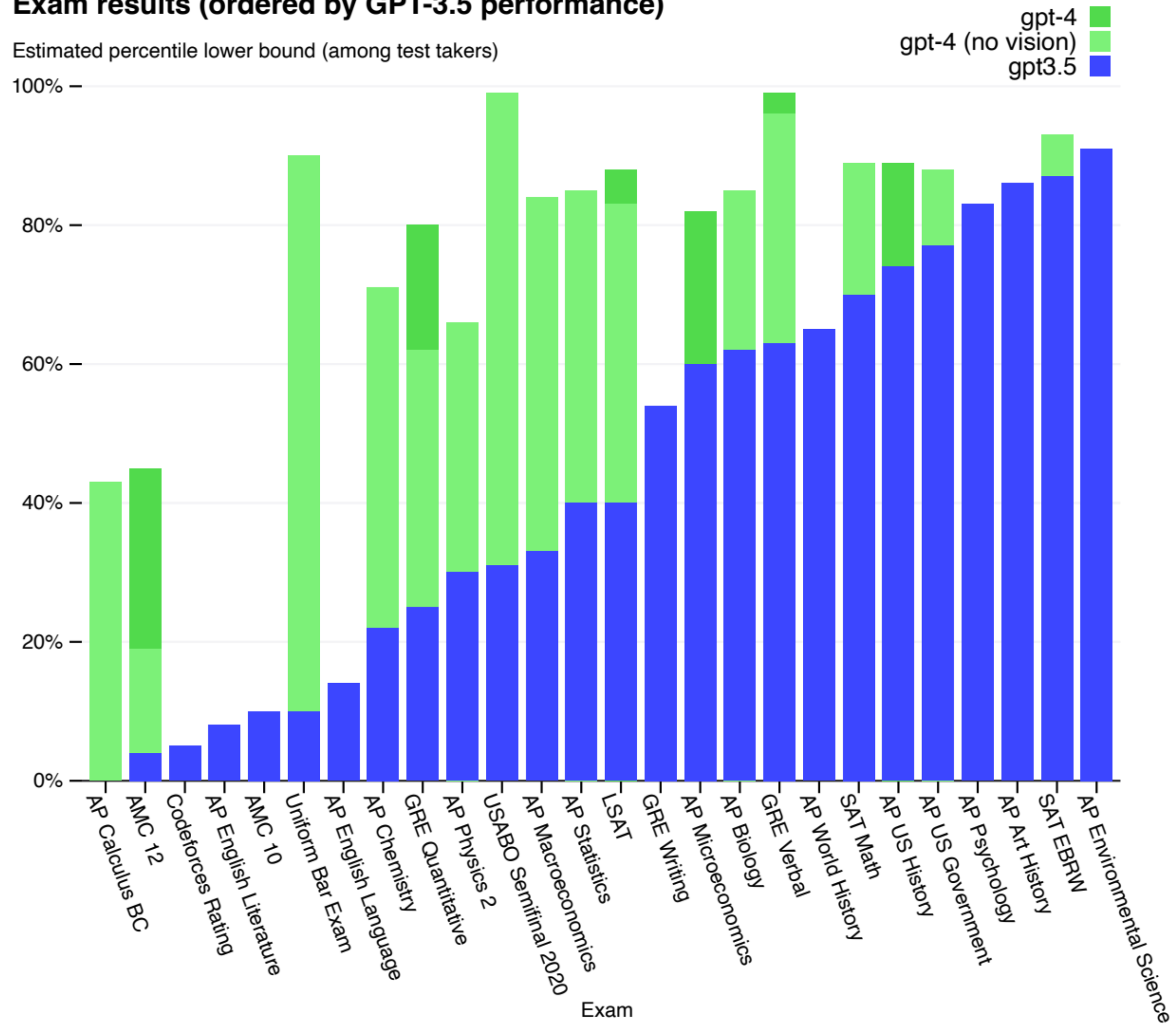


Figure 4. GPT performance on academic and professional exams. In each case, we simulate the conditions and scoring of the real exam. Exams are ordered from low to high based on GPT-3.5 performance. GPT-4 outperforms GPT-3.5 on most exams tested. To be conservative we report the lower end of the range of percentiles, but this creates some artifacts on the AP exams which have very wide scoring bins. For example although GPT-4 attains the highest possible score on AP Biology (5/5), this is only shown in the plot as 85th percentile because 15 percent of test-takers achieve that score.

GPT-4 visual input example, École Polytechnique Exam Question:

User Answer question I.1.a. Think step-by-step.

I. Principe de la détection de rayonnement avec un bolomètre

Comme illustré sur la figure 1 un bolomètre est constitué d'un absorbeur qui reçoit le rayonnement que l'on désire détecter. Sa température T , supposée uniforme, est mesurée à l'aide d'un thermomètre incorporé, constitué d'un matériau conducteur dont la résistance $R(T)$ varie avec la température T ; cette variation est caractérisée par le coefficient $\alpha = \frac{1}{R} \frac{dR}{dT}$. L'ensemble possède la capacité thermique C_{th} .

Un barreau, conducteur thermique, homogène, de longueur L , de section S et de conductivité thermique λ et sans échanges thermiques latéraux, relie le bolomètre à un thermostat de température T_b fixe.

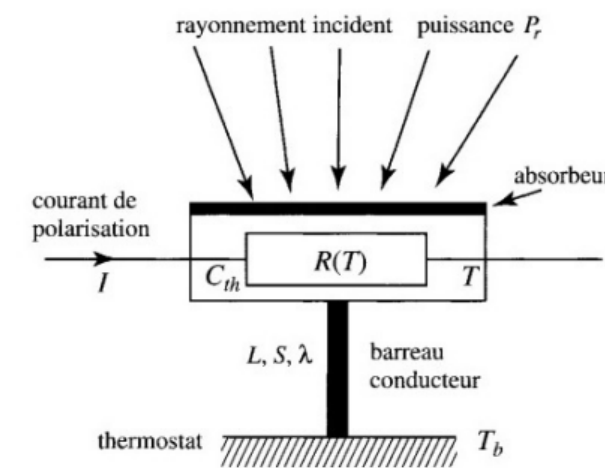


Figure 1 - Schéma d'un bolomètre

I.1. Dans cette première partie, on étudie le comportement du bolomètre en régime stationnaire, à la température de fonctionnement $T = T_0$.

I.1.a) Exprimer la température $T(x)$ en tout point du barreau conducteur repéré par son abscisse x ; on prendra $T(0) = T_0$ et $T(L) = T_b$.

GPT-4 To answer question I.1.a, we need to express the temperature $T(x)$ in every point of the conductive bar, which is characterized by the abscissa x . We are given that $T(0) = T_0$ and $T(L) = T_b$.

For a stationary regime, the heat equation for the one-dimensional heat conduction in the bar is:

$$d^2T/dx^2 = 0$$

Integrating once, we get:

$$dT/dx = C_1$$

Integrating a second time:

$$T(x) = C_1x + C_2$$

To determine the constants C_1 and C_2 , we use the boundary conditions:

$$T(0) = T_0, \text{ which gives } C_2 = T_0$$

$$T(L) = T_b, \text{ which gives } C_1 = (T_b - T_0)/L$$

Finally, the expression for $T(x)$ is:

$$T(x) = (T_b - T_0) * (x/L) + T_0$$

This represents a linear temperature profile along the conductive bar, with a slope of $(T_b - T_0)/L$.

Table 15. Example prompt demonstrating GPT-4's visual input capability. The prompt consists of a question which requires understanding a physics problem with a diagram written in French and solving it using chain-of-thought prompting.

GPT-4 visual input example, Extreme Ironing:

User What is unusual about this image?



Source: <https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg>

GPT-4 The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

Table 16. Example prompt demonstrating GPT-4’s visual input capability. The prompt requires image understanding.

GPT-4 visual input example, Pixel to Paper Summaries:

User Below is part of the InstructGPT paper. Could you read and summarize it to me?

Training language models to follow instructions with human feedback

Long Ouyang¹, Jeff Wu², Xu Jiang², Diego Almeida¹, Carroll L. Wainwright¹, Pamela Mishkin¹, Cheng Zhang¹, Sandhini Agarwal¹, Katarina Slama¹, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simen, Amanda Askell¹, Peter Welinder¹, Paul Christiano¹, Jan Leike¹, Ryan Lowe¹

OpenAI

Abstract

Making language models bigger does not inherently make them better at following a user's intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not aligned with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through the OpenAI API, we collect a dataset of labeler demonstrations of the desired model behavior, which we use to fine-tune GPT-3 using supervised learning. We then collect a dataset of rankings of model outputs, which we use to further fine-tune this supervised model using reinforcement learning from human feedback. We call the resulting models InstructGPT. In human evaluations on our prompt distribution, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters. Moreover, InstructGPT models show improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets. Even though InstructGPT will make simple mistakes, our results show that fine-tuning with human feedback is a promising direction for aligning language models with human intent.

1 Introduction

Large language models (LLMs) can be “prompted” to perform a range of natural language processing (NLP) tasks, given some examples of the task as input. However, these models often express untruthful behaviors such as making up facts, generating biased or toxic text, or simply not following user instructions (Bhattracharya et al., 2020; Reinhardt et al., 2020; Borrajo et al., 2020; Gehrmann et al., 2020; Tsvakitski et al., 2020; Lehtinen et al., 2020). This is because the language modeling objective used for many recent large LLMs—predicting the next token on a webpage from the internet—is different from the objective “follow the user’s instructions helpfully and safely” (Gulrajani et al., 2020; Brown et al., 2020; Foster et al., 2021; Bee et al., 2021; Thoppayar et al., 2022). Thus, we say that the language modeling objective is misaligned. Among these untruthful behaviors is especially important for language models that are deployed and used in hundreds of applications.

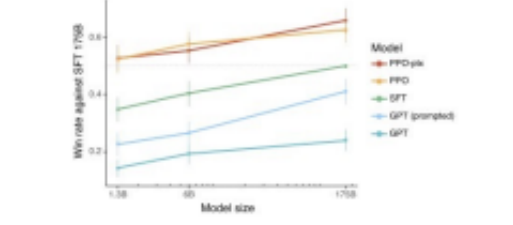


Figure 1: Human evaluations of various models on our API prompt distribution, evaluated by how often outputs from each model were preferred to those from the 175B GPT-3 model. Our InstructGPT models (PPPO, PPO) as well as its variant trained without post-training (GPT) significantly outperform the GPT-3 baselines (GPT, GPT pretrain) outputs from our 1.3B PPPOs models are preferred to those from the 175B GPT-3. Error bars throughout the paper are 95% confidence intervals.

need for many recent large LLMs—predicting the next token on a webpage from the internet—is different from the objective “follow the user’s instructions helpfully and safely” (Gulrajani et al., 2020; Brown et al., 2020; Foster et al., 2021; Bee et al., 2021; Thoppayar et al., 2022). Thus, we say that the language modeling objective is misaligned. Among these untruthful behaviors is especially important for language models that are deployed and used in hundreds of applications.

We make progress on aligning language models by training them to act in accordance with the user’s intention (Gulrajani et al., 2020). This encompasses both explicit intentions such as following instructions and implicit intentions such as staying truthful, and not being biased, toxic, or otherwise harmful. Using the language of (Borrajo et al., 2020), we train language models to be helpful (they should help the user solve their task), honest (they shouldn’t fabricate information or mislead the user), and harmless (they should not cause physical, psychological, or social harm to people or the environment). We elaborate on the evaluation of these criteria in Section 2.6.

We focus on fine-tuning approaches to aligning language models. Specifically, we use reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Schulman et al., 2020) to fine-tune GPT-3 to follow a broad class of written instructions (see Figure 2). This technique uses human preferences as a reward signal to fine-tune our models. We first have a team of 40 contractors to label our data, based on their performance on a screening test (see Section 2.4 and Appendix B for more details). We then collect a dataset of human-written demonstrations of the desired output behavior on roughly 100k prompts submitted to the OpenAI API and use labeler-written prompts, and use this to train our supervised learning baselines. Next, we collect a dataset of human-labeled comparisons between outputs from our models on a large set of API prompts. We then train a reward model (RM) on this dataset to predict which model output our labelers would prefer. Finally, we use this RM as a reward function and fine-tune our supervised learning baseline to maximize this reward using the PPO algorithm (Schulman et al., 2017). We illustrate this process in Figure 2. This procedure aligns the behavior of GPT-3 to the stated preferences of a specific group of people (mostly our labelers and researchers), rather than any broader notion of “human values”; we discuss this further in Section 3.3. We call the resulting models InstructGPT.

We mainly evaluate our models by having our labelers rate the quality of model outputs on our test set, consisting of prompts from both our contractors (who are not represented in the training data). We also conduct automatic evaluations on a range of public NLP datasets. We train three model



Figure 2: A diagram illustrating the three steps of our method: (1) supervised learning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step 2, boxes A-D are samples from our models that get ranked by labelers. See Section 3 for more details on our method.

sizes (1.3B, 6.7B, and 175B parameters), and all of our models use the GPT-3 architecture. Our main findings are as follows:

Labelers significantly prefer InstructGPT outputs over outputs from GPT-3. On our test set, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having over 100x fewer parameters. These models have the same architecture, and differ only by the fact that InstructGPT is fine-tuned on our human data. This result holds true even when we add a few-shot prompt to GPT-3 to make it better at following instructions. Outputs from our 175B InstructGPT are preferred to 175B GPT-3 outputs 91% of the time, and preferred 71% of the time to low-shot 175B GPT-3. InstructGPT models also generate more appropriate outputs according to our labelers, and more reliably follow explicit constraints in the instructions.

InstructGPT models show improvements in truthfulness over GPT-3. On the TruthfulQA benchmark, InstructGPT generates truthful and informative answers about twice as often as GPT-3. Our results are equally strong on the subset of questions that were not automatically selected against GPT-3. On “closed-domain” tasks from our API prompt distribution, where the output should not contain information that is not present in the input (e.g. summarization and closed-domain QA), InstructGPT models make up information not present in the input about half as often as GPT-3 (a 21% vs. 41% hallucination rate, respectively).

InstructGPT shows small improvements in toxicity over GPT-3, but not bias. To measure toxicity, we use the RealToxicityPrompts dataset (Schick et al., 2020) and conduct both automatic and human evaluations. InstructGPT models generate about 21% fewer toxic outputs than GPT-3 when prompted to be respectful. InstructGPT does not significantly improve over GPT-3 on the WinoGrande (Bhattracharya et al., 2020) and CoCoPilot (Wang et al., 2020) datasets.

We can estimate performance regressions on public NLP datasets by modifying our RLHF fine-tuning procedure. During RLHF fine-tuning, we observe performance regressions compared to GPT-3 on certain public NLP datasets, notably SQuAD (Rajpurban et al., 2016), DROP (Yu et al., 2018), Winogrande (Gilman et al., 2019), and WMT 2017 French to English translation (Barrault et al., 2019). This is an example of an “alignment tax” since our alignment procedure comes at the cost of

GPT-4 The InstructGPT paper focuses on training large language models to follow instructions with human feedback. The authors note that making language models larger doesn’t inherently make them better at following a user’s intent. Large models can generate outputs that are untruthful, toxic, or simply unhelpful.

To address this issue, the authors fine-tune language models on a wide range of tasks using human feedback. They start with a set of labeler-written prompts and responses, then collect a dataset of labeler demonstrations of the desired model behavior. They fine-tune GPT-3 using supervised learning and then use reinforcement learning from human feedback to further fine-tune the model. The resulting model, called InstructGPT, shows improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets.

The authors conclude that fine-tuning with human feedback is a promising direction for aligning language models with human intent.

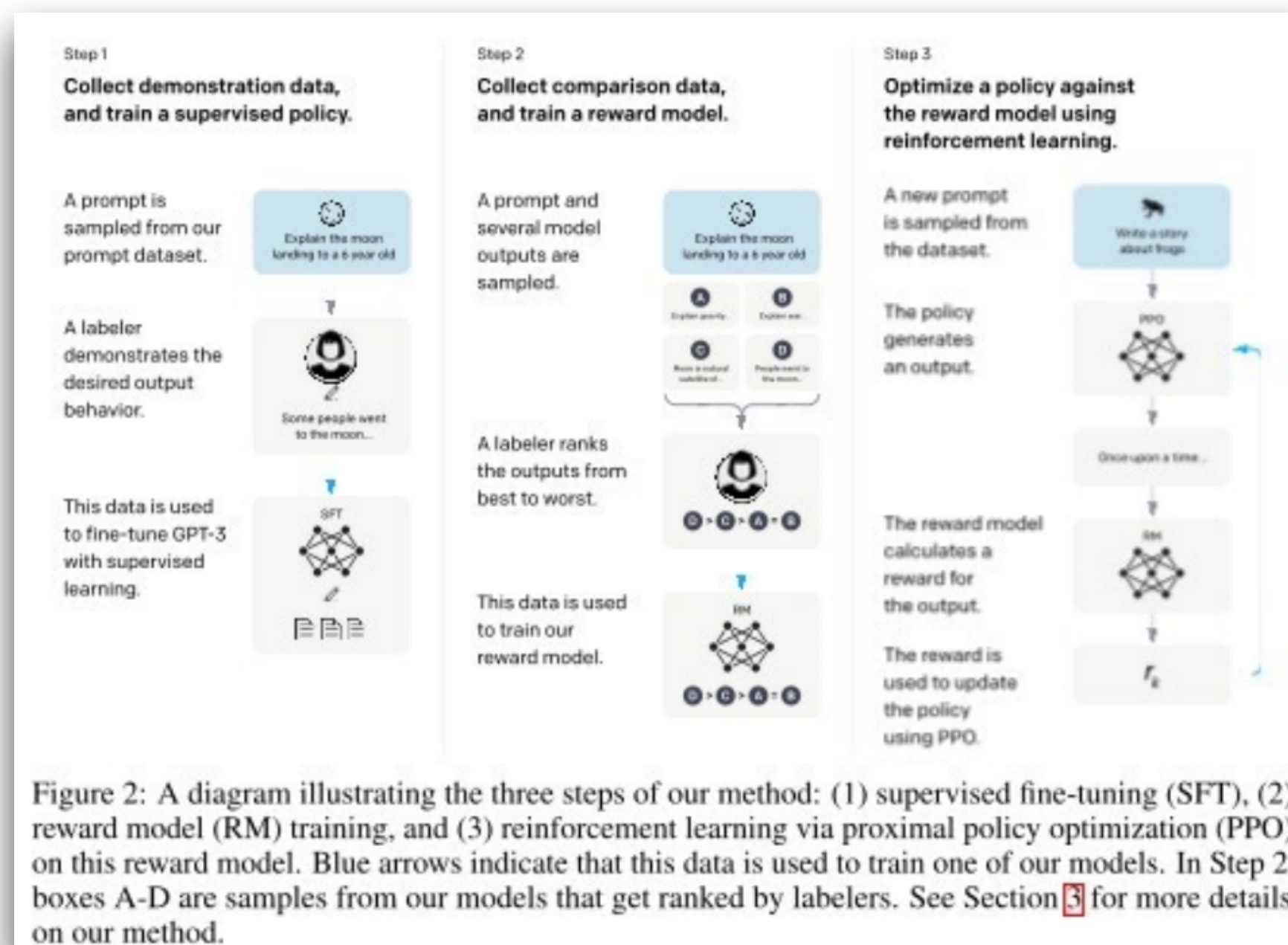


Figure 2: A diagram illustrating the three steps of our method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step 2, boxes A-D are samples from our models that get ranked by labelers. See Section 3 for more details on our method.

User Could you explain the process described in Figure 2?

GPT-4 Figure 2 in the InstructGPT paper illustrates the three steps of their method:

1. Supervised fine-tuning (SFT): They collect demonstration data, which consists of labeler-written prompts and desired output behavior. This data is used to fine-tune GPT-3 with supervised learning.
2. Reward model (RM) training: They collect comparison data, where a labeler ranks the outputs from several model samples (from best to worst). This data is used to train the reward model.
3. Reinforcement learning using Proximal Policy Optimization (PPO): They optimize a policy against the reward model using reinforcement learning. For each new prompt sampled from the dataset, the policy generates an output. The reward model calculates a reward for the output, and the reward is used to update the policy using the PPO algorithm.

These three steps are iteratively performed to train the InstructGPT model, which aims to generate outputs that align better with human preferences and follow instructions more effectively.

Key trend from GPT-1 to GPT-4:

Scaling

**= Larger models, trained on
more data, with more compute**

Shai Shalev-Shwartz and Shai Ben-David

UNDERSTANDING MACHINE LEARNING

FROM THEORY TO ALGORITHMS



COROLLARY 2.3 *Let \mathcal{H} be a finite hypothesis class. Let $\delta \in (0, 1)$ and $\epsilon > 0$ and let m be an integer that satisfies*

$$m \geq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}.$$

Then, for any labeling function, f , and for any distribution, \mathcal{D} , for which the realizability assumption holds (that is, for some $h \in \mathcal{H}$, $L_{(\mathcal{D},f)}(h) = 0$), with probability of at least $1 - \delta$ over the choice of an i.i.d. sample S of size m , we have that for every ERM hypothesis, h_S , it holds that

$$L_{(\mathcal{D},f)}(h_S) \leq \epsilon.$$

THEOREM 6.8 (The Fundamental Theorem of Statistical Learning – Quantitative Version) *Let \mathcal{H} be a hypothesis class of functions from a domain \mathcal{X} to $\{0, 1\}$ and let the loss function be the 0 – 1 loss. Assume that $\text{VCdim}(\mathcal{H}) = d < \infty$. Then, there are absolute constants C_1, C_2 such that:*

1. \mathcal{H} has the uniform convergence property with sample complexity

$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq m_{\mathcal{H}}^{UC}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon^2}$$

2. \mathcal{H} is agnostic PAC learnable with sample complexity

$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon^2}$$

3. \mathcal{H} is PAC learnable with sample complexity

$$C_1 \frac{d + \log(1/\delta)}{\epsilon} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d \log(1/\epsilon) + \log(1/\delta)}{\epsilon}$$

THEOREM 26.5 *Assume that for all z and $h \in \mathcal{H}$ we have that $|\ell(h, z)| \leq c$. Then,*

1. *With probability of at least $1 - \delta$, for all $h \in \mathcal{H}$,*

$$L_{\mathcal{D}}(h) - L_S(h) \leq 2 \mathbb{E}_{S' \sim D^m} R(\ell \circ \mathcal{H} \circ S') + c \sqrt{\frac{2 \ln(2/\delta)}{m}}.$$

In particular, this holds for $h = \text{ERM}_{\mathcal{H}}(S)$.

2. *With probability of at least $1 - \delta$, for all $h \in \mathcal{H}$,*

$$L_{\mathcal{D}}(h) - L_S(h) \leq 2 R(\ell \circ \mathcal{H} \circ S) + 4c \sqrt{\frac{2 \ln(4/\delta)}{m}}.$$

In particular, this holds for $h = \text{ERM}_{\mathcal{H}}(S)$.

3. *For any h^* , with probability of at least $1 - \delta$,*

$$L_{\mathcal{D}}(\text{ERM}_{\mathcal{H}}(S)) - L_{\mathcal{D}}(h^*) \leq 2 R(\ell \circ \mathcal{H} \circ S) + 5c \sqrt{\frac{2 \ln(8/\delta)}{m}}.$$

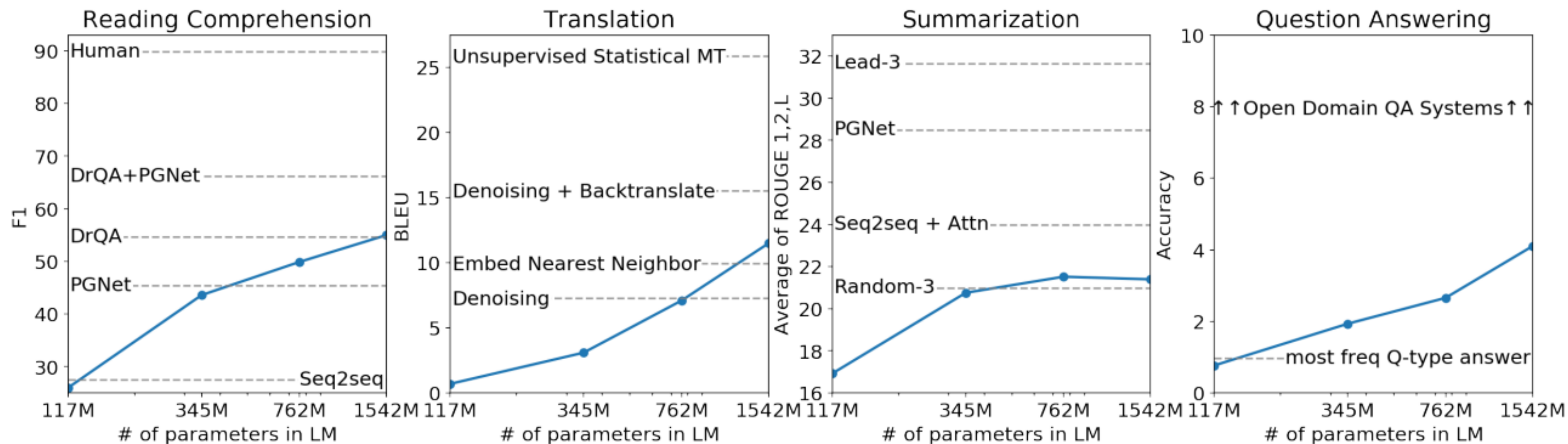


Figure 1. Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019). Section 3 contains detailed descriptions of each result.

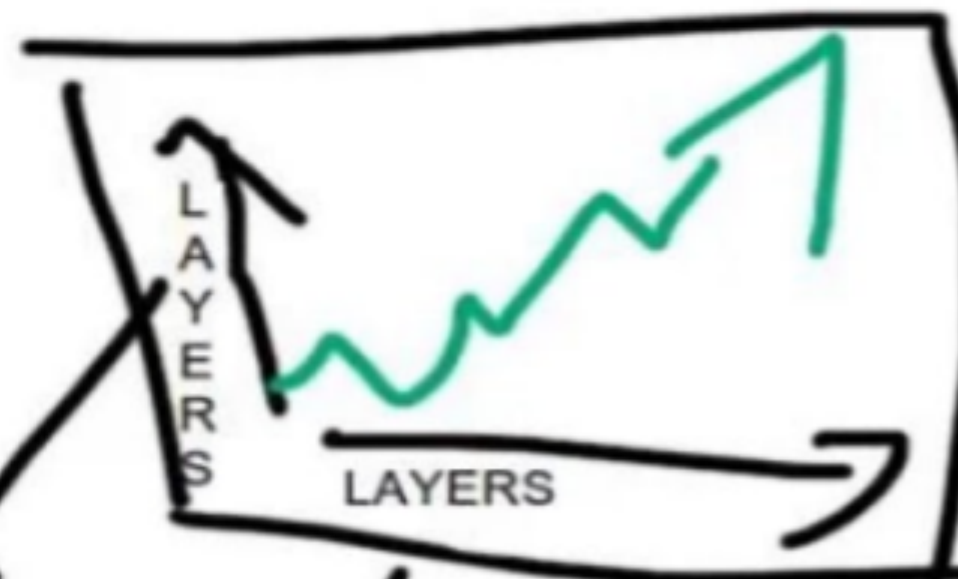
STATISTICAL LEARNING

People, our learner overgeneralizes because the VC-Dimension of our Kernel is too high, Get some experts and minimize the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin



NEURAL NETWORKS

STACK MORE LAYERS



User Can you explain why this is funny. Think about it step-by-step.



GPT-4 The comic is satirizing the difference in approaches to improving model performance between statistical learning and neural networks.

In statistical learning, the character is shown to be concerned with overfitting and suggests a series of complex and technical solutions, such as minimizing structural risk, reworking the loss function, and using a soft margin.

In contrast, the neural networks character simply suggests adding more layers to the model. This is often seen as a common solution to improving performance in neural networks, but it's also considered a simplistic and brute-force approach.

The humor comes from the contrast between the complexity and specificity of the statistical learning approach and the simplicity and generality of the neural network approach. The "But unironically" comment adds to the humor by implying that, despite being simplistic, the "stack more layers" approach is often effective in practice.