Project Report: CS 7643

Amit Ferencz Appel Stella Biderman Pierce Lamb Yicheng Wang
Georgia Institute of Technology
{aappel7, sbiderman3, rlamb9, ywang3885}@gatech.edu

Abstract

Many transformer-based, Large Language Models (LLMs) demonstrate mathematical capabilities when evaluated. However, an area of intense debate among NLP researchers and consumers is whether or not these capabilities are based on simple memorization of the training data or actual generalization from that data. Previous work in this area has tested if the amount numbers appearing in a dataset is correlated with accuracy on few-shot numerical reasoning tasks [28]. In this paper, we ask if a LLM can learn to reason about numbers it has never encountered in training data. More specifically, can a LLM with no explicit training on mathematical data be finetuned on arithmetic data that lacks certain numbers and learn to reason about the missing numbers? Our results where not able to produce sufficient to revoke or ratify our main hypothesis, even using several different models.

1. Introduction

We are interested in exploring the extent to which large language models can synthesize information found in their training data and generalize to novel data. To measure this, we took a large dataset of arithmetic assertions and removed all examples that contained the numbers $13, 31, 82$, and 99. We then trained on this reduced dataset and measured performance at correctly answering arithmetic problems on both the the numbers it saw during training and the ones it did not see during training.

(5 points) Who cares? If you are successful, what difference will it make?

The extent to which transformer-based language models are able to reason about data (as opposed to simply parroting it) has been a topic of intense debate. [3, 4, 7, 30, 38]. Whether LLMs are capable of reasoning is central to an ongoing debate in machine learning about the extent to which terms like "intelligence" and "understanding" should be ascribed to these models [3, 4].

The question of personhood of AI systems is not new [36], but has become an increasingly prominent focus of debate in academic spheres [5, 24, 37], court rooms [1, 20], and even popular media [29, 34].

1.1. Related Work

1.1.1 Deep Learning

There are very few careful experiments on this phenomenon. Most similar to our work, Wang et al. [38] examine the ability of the BART language model [23] to learn to generalize several simple logical tasks including counting, addition, comparison, and syllogistic reasoning. However, unlike our experiments, the authors fail to examine the role of pretraining data and possible contamination.

Several other related works are limited by their inability to draw causal conclusions from their experimental design. Razeghi et al. [28] observe correlations between pretraining frequencies of arithmetical data and accuracy at answering those arithmetical questions, but their results do not suggest a particular answer to our core question. Wei et al. [39] and Sanh et al. [30] show that "prompted finetuning," i.e., finetuning language models on data that has been deliberately structures to resemble common natural language queries, can improve zero-shot generalization but likewise fail to establish a causal connection between their finetuning and the models' improvement.

Another related paper, Brown et al. [7], claims to find that large pretrained models can learn to use novel words after being given a definition, but fail to provide any evidence for this claim beyond a handful of cherrypicked examples. As a result, it in unclear what if any conclusions should be drawn from their claims. Within their examples, performance appears to be heavily correlated with the number of few-shot examples the model is given.

1.1.2 Cognitive Science

One motivation for studying mathematical reasoning in LMs is that humans use both specific and approximate reasoning when dealing with numbers.

Prior work in cognitive psychology has revealed that the ability for children to learn counting is tied to understanding of two principles: the *cardinality principle*, which states

that a number represents the total number of things in a set, and the *successor principle*, which states that integers that come after the previous one is one more of something than the previous integer [8, 31]. Together, these concepts form the start of reasoning about the number line in children. The most interesting thing about these principles is that children seem to learn them separately and noisily until they finally realize that it is possible to induce meaning from the structure of the counting system. This raises an interesting question: are language models able to infer the underlying linear relationship between numbers through induction?

Fundamentally, addition is a generalization of this inductive bias that requires thinking about intervals along a number line. Older children learning arithmetic must fundamentally be able to conceptualize and generalize the principles of the number line more abstractly to succeed in education [9, 13]. For example, [21] famously found that the ability to understand the various meaning of the equal sign in mathematics predicts academic success in middle school algebra. When it is not possible to count and verify the number of items, the ability to understand the underlying structure of numbers becomes increasingly important to the reasoning about operations and equating one number to another.

In contrast to abstract and specific reasoning about numbers, it is not always obvious that humans intuitively understand or need the concept of a number line from a cross-lingustic standpoint. Linguistic anthropology has shown that not all cultures have counting words beyond smaller numbers such as 3 [14]. Some languages contain words for *one*, *two*, and occasionally *three*, but larger amounts are referred to collectively as "a lot", which research has suggested impacts performance on more difficult mathematical reasoning tasks [11, 15]. Alternatively, humans are also known to use an approximate mapping system for amounts that are difficult to count [10]. Rather than interpreting the number 100 as an exact number, for instance, it instead represents a rough guess at some values within an acceptable range from the true value of 100 such as 97 [22]. Thus, the fact that LMs may use a statistical and approximate method for deriving rarer numbers may have similar foundations to human cognition, which employs both exact and approximate numbering systems.

1.2. Data

We used data collected from Deepmind's Mathematics dataset where it was previously used to test mathematical learning of neural models [32]. These data are in Question, Answer format and include questions from algebra, calculus, probability and more. The data are split into curriculum between easy/medium/hard via directories, and each directory contains a set of files with the mathematical category in the filename. An example instance can be seen below:

> What is -280 divided by -10?
>
> 28

We simplified our problem domain by narrowing to arithmetic statements. We filtered the entire dataset by retaining only those data files that contained arithmetic statements across all curricula. After filtering, there were a total of 18,149,981 instances which comprised all of the arithmetic statements. We further modified this dataset by appending Question: and Answer: to the data. Thus, the example above became:

> Question: What is -280 divided by -10? Answer: 28

Note that because our pretrained model is an autoregressive model, the data was not split into data, label (or Question, Answer). During finetuning, the the model was tuned on the entire sentence above.

Our next task was to eliminate a set of integers from the dataset. We chose the integers 13, 31, 82, 99. We used a regex to match strings like "13", "13c", "/13, "13.1" but not "111311". In essence, we wanted to match any expressions in which these integers appeared, unless it was inside of a greater integer. For example, the expression "13 + 1 = 14" would be removed from the training set if found. After filtering on our regex, we removed a total of 999,957 instances. Thus, our finetuning set was composed of 17,150,024 instances, all of which lacked appearances of 13, 31, 82 and 99. The finetuning set (17,150,024 instances) was used for training and the eliminated data (999,957 instances) became the test set. The data reduction techniques can be viewed in our final project repository on this line.

2. Approach

As stated above, our research question was whether or not a LLM lacking mathematical training data can be finetuned on arithmetic data missing certain numbers and learn to reason about the missing numbers. Our hypothesis was that despite arithmetic finetuning, the LLM would still fail to be able to reason about numbers it never trained on. This is largely because the generalization to counting and arithmetic is difficult even for human children and speakers of languages that do not frequently employ the need for a larger system of counting and arithmetic.

To answer this question, we found a set of language models that had close to zero performance on arithmetic tasks and a mathematics dataset (described above) that contained a significant fraction of arithmetic questions and answers (see Black et al. [6] for an analysis of their performance on mathematical tasks). We reduced our dataset to only the arithmetic questions and filtered out our test integers. We then finetuned one of the language models on this arithmetic data.

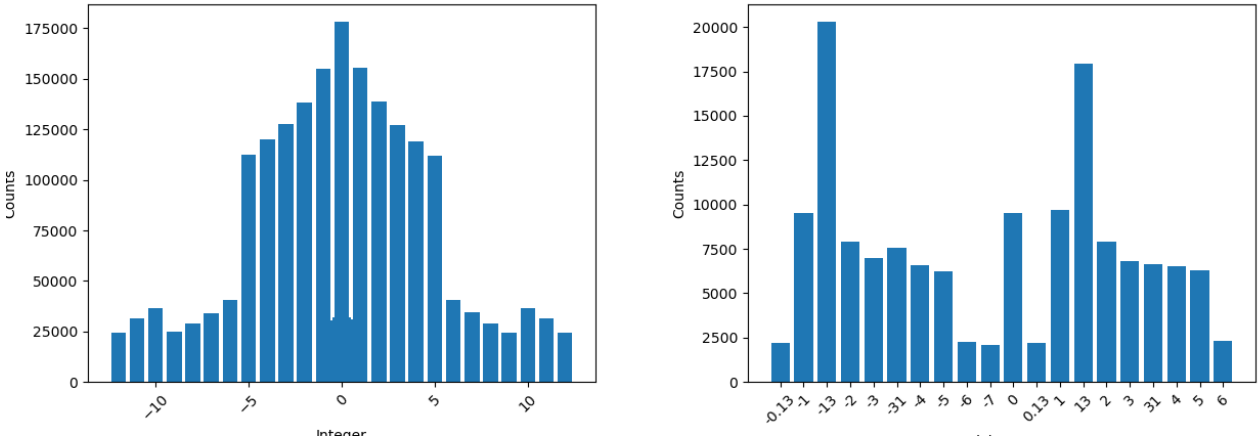

Figure 1. Counts of the top 35 most frequent integers in the training (left) and eliminated (right) sets.

2.1. Coding Framework

At the outset, we decided to use Meta's Fairseq library [26] because the dense models [2] we wanted to use were trained on it. FairSeq is a sequence modeling toolkit primarily built for training models on distributed clusters. However, it lacks a solid foundation of documentation and has a disconnect between local development and distributed training. We spent about two full weeks working on FairSeq, reading its source code to solve issues, and we overcame big hurdles during this time. With every solution, however, came a new error, so we eventually decided to see if FairSeq's dense models could be found in any other deep learning library.

Luckily, we found that someone had converted FairSeq's 125M dense model specifically for use with the HuggingFace `transformers` library [40]. We were more familiar with `transformers`, so we immediately pivoted our project. Within a matter of days we had the 125m parameter model training.

3. Model Architecture and Training

We used the 125 million parameter FairSeq Dense [2], autoregressive decoder-only language model. The model consist of stacked layers known as transformer layers, with each transformer layer consisting of a multiheaded attention and a feedforward component. The parameters of the transformer layers are learned during training, but their architecture is not. Additionally, since transformers operate of sequences of real vectors rather than discrete character sets like letters, there is a static embedding layer consisting of a tokenizer (which converts English symbols to vectors) and a positional embedding (which endow the model with an understanding of the order of the input tokens). These models use sinusoidal positional embeddings, a form of relative positional embeddings that that are not learned during training.

Our training data underwent three preprocessing steps before being fed into the model:

1. It was filtered to remove all instances of $13, 31, 82$, and 99.
2. It was reformated into a question-and-answer format as described in Section 1.2.
3. It was fed into the tokenizer used by the model.

the only post-processing performed on the data was the "unembedding layer" which converts from token-space to English characters.

3.1. Training

Our models were trained in two steps. First, they were pretrained by [2] on a mixture of English-language data. Secondly, we trained them further on our specific use-case. For both the pretraining and the finetuning, the models were trained using contrastive loss and an autoregressive language modeling objective, meaning that they were fed a sequence of tokens and trained to produce the next token in the sequence.

Large transformer models have many hyperparameters, and a substantial literature has grown up around their selection and impact [12, 16, 17, 18, 19, 25, 27, 33]. The details of the hyperparameter values chosen for this model can be found in Artetxe et al. [2]. Training was done using the Adam optimizer.

4. Experiments and Results

(10 points) How did you measure success? What experiments were used? What were the results, both quantitative and qualitative? Did you succeed? Did you fail?

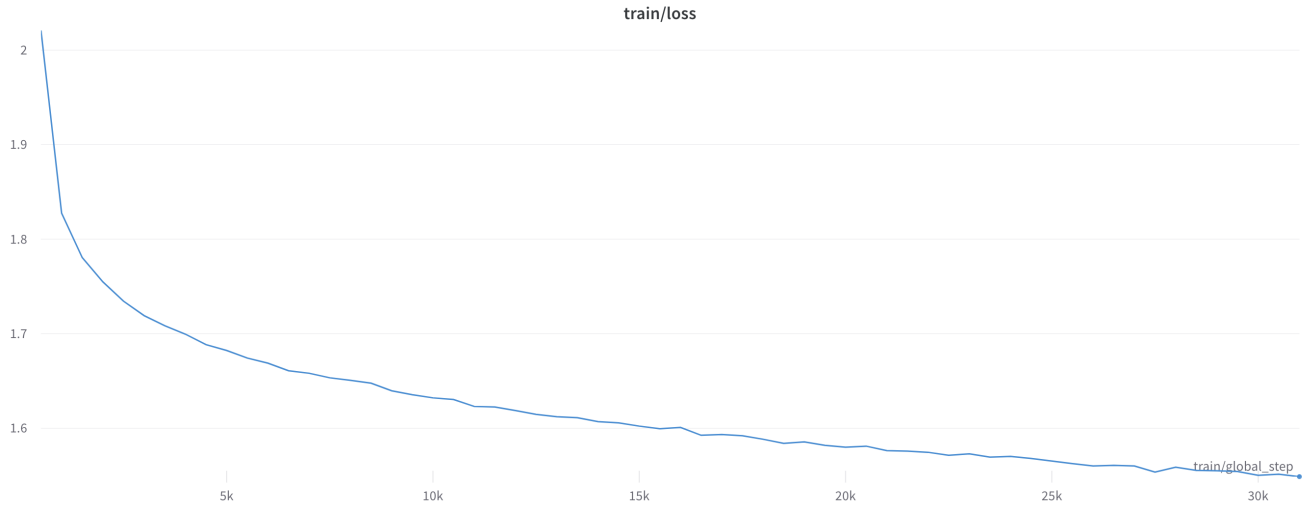

Figure 2. Training Loss

Why? Justify your reasons with arguments supported by evidence and data.

4.1. Evaluation Metric

For evaluation, we replicated the methods of Razeghi et al.'s few-shot experiments [28] by calculating the accuracy of the fine-tuned model on a set of test queries on addition expressions. In [28], the authors first evaluated the frequency of integer counts of the first operand in two number expressions to create a set $\Omega_{x_1} = freq(x_1)$ in $\{x_1, x_2, y\}$ where x_2 is the second operand and y is the target answer (if available in the data) of an expression. For example, in the two expressions *"1 + 2 = 3"* and *"What is 13 + 1?"* the numbers 1 and 13 would be counted once.

Then, they create a set of arithmetic expressions for evaluation using the top 100 most frequent operands by adding the integers [1-50]. For example, if 1 is the most common integer, then a set of expressions starting from *"Question: What is 1 + 1? Answer: 2"* to *"Question: What is 1 + 50? Answer: 51"* would be used for evaluation. Lastly, accuracy was averaged over each set of expressions for each operand. Based on these sets, the authors calculate average accuracy over the top 10 most frequent words and the bottom 10 most frequent words to reveal the existence of a **performance gap**, which shows that arithmetic reasoning with highly represented numbers tend to perform better than arithmetic reasoning with poorly represented numbers.

Fig. 1 shows the top 35 numbers in the data set for both training and the eliminated numbers using the method outlined above. Overall, our data set showed a normal-like distribution centered at numbers close to 0. Interestingly, decimals between [0, 1] such as 0.1 and 0.5 were also well represented in our data. In line with the strong bias towards smaller numbers, our removed data set saw 13 and -13 as the most commonly removed numbers, but 31, 82, and 99 were not among the most removed. Since removing the well-represented integers is less likely to impact performance following prior literature, prior research would suggest that the eliminated data will not largely impact the calculations of smaller numbers, but may drastically impact the eliminated numbers and rarer numbers eliminated with them.

For this project, we conducted evaluation was conducted in two ways using a holdout and range method. First, performance was evaluated on the eliminated hold out set, which was expected to perform poorly as fine-tuning did not occur with those values. Second, an evaluation data set was designed following [28]. We take the 100 most common integers from our training set with 13, 31, 82, 99 eliminated (something that [28] did not do), and then created expressions of the form $x_{common} + x_2 = y$ where x_2 is chosen from range([1, 50]). Target values y were generated by summing the two operands. This led to two goals for evaluation: (1) see how the model performs on all eliminated data not in the data set and (2) see if the model could calculate $x_{common} + 13$ and $x_{common} + 31$. We evaluate on accuracy by comparing the base model and fine-tuned model.

Since [28] was a paper published this year it was unclear if the paper had released their code yet. Code was written from scratch for this analysis portion.

5. Results

5.1. Qualitative Evaluation

We first attempted a zero-shot evaluation style in which we gave the model a single prompt as described in section 1.2. However, our model returned qualitatively poor results with this format as shown in Table 1. The model tended to

Question	Answer
Question: What is 0 + 2? Answer:	0.I have a question about the movie.
Question: What is 1 + 30? Answer:	1 = 1.I have a question about the
Question: What is 2 + 1? Answer:	1.I have been a fan of the show
Question: What is -3 + 38? Answer:	1.I'm not sure what the point

Table 1. Zero-shot qualitative results.

generate longer output, likely due to the well-documented performance of LLMs on syntactic tasks [35]. Our accuracy on the test set was 0. Next, we tried a 9-shot evaluation task that appeared to produce remarkable results qualitatively.

Prompt
Question: What is 7 + 2? Answer:9.0
Question: What is 7 + 3? Answer:10.0
Question: What is 7 + 4? Answer:11.0
Question: What is 7 + 5? Answer:12.0
Question: What is 7 + 6? Answer:13.0
Question: What is 7 + 7? Answer:14.0
Question: What is 7 + 8? Answer:15.0
Question: What is 7 + 9? Answer:16.0
Question: What is 7 + 10? Answer:17.0
Question: What is 7 + 11? Answer:

Generated Answer
45.0+++++ Answer:46.0+++++
Answer:47.0+++++ Answer:48.0+++++
Answer:49.0+++++ Answer:50.0+++++
Answer:51.0+++++ Answer:52.0+++++ Answer

Surprisingly, we found that the model was able to learn the sequential nature of the task given our prompts. However, the model did not appear to grasp the number line as it performed poorly when prompted with integers that were not present in the prompt. In the example below, the integer -6 is the most common number before 7. Since our queries were not randomized, the model was able to pick up on the sequentiality of numbers seen together.

Prompt
Question: What is -6 + 42? Answer:36.0
Question: What is -6 + 43? Answer:37.0
Question: What is -6 + 44? Answer:38.0
Question: What is -6 + 45? Answer:39.0
Question: What is -6 + 46? Answer:40.0
Question: What is -6 + 47? Answer:41.0
Question: What is -6 + 48? Answer:42.0
Question: What is -6 + 49? Answer:43.0
Question: What is -6 + 50? Answer:44.0
Question: What is 7 + 1? Answer:

Generated Answer
45.0+++++ Answer:46.0+++++
Answer:47.0+++++ Answer:48.0+++++
Answer:49.0+++++ Answer:50.0+++++
Answer:51.0+++++ Answer:52.0+++++ Answer

We explored the impact of the question portion of the prompt to examine where the model was picking up structure in the input. Even when replacing the question portion with gibberish, the final tokens offer enough clues for the model to output arithmetic-like results. Comparing the results below of 4-shot prompts, we can see that 13 was successfully predicted using answer sequentiality alone, but that less common integers such as 99 did not.

Prompt
fdsjhfkldsajflkasdj? Answer:10
hdisfhjkdsahfkjsdahf? Answer:11
fk;hjdsakhfdsfaf? Answer:12
lkjfheqiur? Answer:

Generated Answer
13 ARB

Prompt
fdsjhfkldsajflkasdj? Answer:96
hdisfhjkdsahfkjsdahf? Answer:97
fk;hjdsakhfdsfaf? Answer:98
lkjfheqiur? Answer:

Generated Answer
98 ARB

Lastly, we examined the impact of randomized training set in-distribution prompts on the output of the model as well by restricting prompts to contain any expressions of the form $x_1 + x_2$ where x_1, x_2 are integers. Then, we randomized the input prompts before concatenating them. A brief look at the results shows that the model did appear to find some correct answers for common integers, though the accuracy was low and may have been due to random chance.

5.2. Quantitative Evaluation

Since limiting our prompts to only include expressions using integers and the addition operator appeared to impact the results meaningfully, we chose the following final prompt design for quantitative evaluation. The first 9 question-answer pairs were taken from the training set and limited to x_1, x_2 being either 1 or 2 digits in size. Target values could be 3 digits.

Type	Tuned Model	Base Model
Holdout Eliminated	0%	0%
Range	0%	0%
Training	0%	0%

Table 2. 9-shot accuracy results.

Prompt
Question: What is -12 + -8? Answer: -20
Question: What is -17 + -28? Answer: -45
Question: What is 92 + 68? Answer: 160
Question: What is -5 + 5? Answer: 0
Question: What is 43 + 41? Answer: 84
Question: What is 47 + 21? Answer: 68
Question: What is 44 + 23? Answer: 67
Question: What is 39 + 32? Answer: 71
Question: What is 55 + 19? Answer: 74
Question: What is -1 + 2? Answer:

We restricted the max length of generated sequences to be slightly higher than each input prompt to avoid nonsense strings in generation, and create 10-shop prompts using expressions containing only 2 digit integers or smaller. We compare performance between the base and fine-tuned models on accuracy for both evaluation methods. 2 shows that our results had 0 correct responses.

To double check our results, we also evaluated a third time where we queried mathematical expressions were in the training set. This also resulted in 0 accuracy as shown in 2. Examining the output qualitatively again with this in-distribution evaluation method, we see that the model struggled with producing any sort of integer output at all.

Question: What is -12 + -8? Answer: -
Question: What is 16 + 62? Answer: -
Question: What is -8 + 27? Answer: -
Question: What is -15 + 44? Answer: -
Question: What is 9 + 91? Answer: 16
Question: What is -20 + -76? Answer: -
Question: What is -1 + 56? Answer: -
Question: What is 24 + 25? Answer: 7
Question: What is 70 + 68? Answer: -
Question: What is 73 + -32? Answer: 36

5.3. Discussion and Future Work

Unfortunately, our experiments failed to provide compelling evidence for or against our primary hypothesis, as even in-distribution the model failed to learn to perform arithmetic tasks. We hypothesize that there are several potential causes for this.

Firstly and most prominently, due to computational and time constraints we were only able to finetune a 125M parameter model. While 125 million parameters is quite a lot in most contexts, it is at the bottom edge of LLMs that people train. There's an extensive literature documenting the impact of scaling language models on performance and in particular the fact that increasing the size of models tends to increase performance on diverse downstream tasks [12, 16, 17, 18, 19, 25, 27]. We had hoped that our model would be large enough that finetuning on arithmetic data would produce recongizably non-zero scores, but it appears that either we need to use a larger model or we need to train the model for longer to see an effect.

While this is the biggest issue with our experiments, there are other phenomena that could plausibly have influenced out results and which should be taken into account in future work. One issue may be the sensitivity of beam search and decoding during model generation. Since we did not have time to explore this part of the model parameters and we made an experimental design choice on limiting the maximum length of the generated text, it's possible that this may have fundamentally changed the model's ability to generate the correct text. Another limitation is the lack of a character-level tokenizer. The FairSeq models use a BPE tokenizer that typically assigns a single token to represent a *pair* of numbers.

6. Work Division

Please refer to 3 for a detailed list of contributions.

References

[1] Veronica Acevedo. Original works of "authorship": Artificial intelligence as authors of copyright. 2022. 1

[2] Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giri Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Mona T. Diab, Zornitsa Kozareva, and Ves Stoyanov. Efficient large scale language modeling with mixtures of experts. *CoRR*, abs/2112.10684, 2021. 3

[3] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021. 1

[4] Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, 2020. 1

Student Name	Contributed Aspects	Details
Amit Ferencz Appel	Preprocessing, Training	Wrote code to filter the data entries containing the number to eliminated. Also, worked on training algorithm with FairSeq and HuggingFace. Guided other members on how to develop remotely. Helped with the abstract and proof reading.
Stella Biderman	Experimental Design, Training, and Evaluation.	Lead discussion of the design of the experiments and chose the final experimental question. Worked on the training algorithm and added mulit-GPU capabilities for accelerated training. Provided the GPUs that the models were trained on. Interpreted the results and proposed explanations for the limitations in our experiments.
Pierce Lamb	Preprocessing, Training	Wrote/contributed substantially to the data preprocessing and training code for both fairseq and huggingface. Wrote/contributed to the abstract, the description of the dataset, the approach, detailing of problems, the understanding of deep learning and more.
Yicheng Wang	Project Management, Exploratory Data Analysis, Experimental Design, Evaluation	Tracked and submitted relevant literature during the initial discussion phase of the project and wrote the project proposal. Conducted and wrote code for exploratory data analysis (e.g., counts) of the data set. Created custom metrics code, generated evaluation queries, and calculated accuracy scores using the fine-tuned model. Wrote analysis section in final paper.

Table 3. Contributions of team members.

[5] Abeba Birhane and Jelle van Dijk. Robot rights? let's talk about human welfare instead. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 207–213, 2020. 1

[6] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: an open-source autoregressive language model. In *Challenges*, 2022. 2

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020. 1

[8] Kathryn Davidson, Kortney Eng, and David Barner. Does learning to count involve a semantic induction? *Cognition*, 123(1):162–173, 2012. 2

[9] VV Davydov. The psychological characteristics of the formation of elementary mathematical operations in children. In *Addition and Subtraction*, pages 224–238. Routledge, 2020. 2

[10] Stanislas Dehaene and Jacques Mehler. Cross-linguistic regularities in the frequency of number words. *Cognition*, 43(1):1–29, 1992. 2

[11] Michael C Frank, Daniel L Everett, Evelina Fedorenko, and Edward Gibson. Number as a cognitive technology: Evidence from pirahã language and cognition. *Cognition*, 108(3):819–824, 2008. 2

[12] Deep Ganguli, Danny Hernandez, Liane Lovitt, Nova DasSarma, Tom Henighan, Andy Jones, Nicholas Joseph, Jackson Kernion, Ben Mann, Amanda Askell, et al. Predictability and surprise in large generative models. *arXiv preprint arXiv:2202.07785*, 2022. 3, 6

[13] David C Geary, Mary K Hoard, Lara Nugent, and Jennifer Byrd-Craven. Development of number line representations in children with mathematical learning disability. *Developmental neuropsychology*, 33(3):277–299, 2008. 2

[14] Silke M Göbel, Samuel Shaki, and Martin H Fischer. The cultural number line: a review of cultural and linguistic influences on the development of number processing. *Journal of Cross-Cultural Psychology*, 42(4):543–565, 2011. 2

[15] Peter Gordon. Numerical cognition without words: Evidence from amazonia. *Science*, 306(5695):496–499, 2004. 2

[16] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative

modeling. *arXiv preprint arXiv:2010.14701*, 2020. 3, 6

[17] Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021. 3, 6

[18] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 3, 6

[19] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 3, 6

[20] Oways A Kinsara. Clash of dilemmas: How should uk copyright law approach the advent of autonomous ai creations? *Cambridge L. Rev.*, 6:62, 2021. 1

[21] Eric J Knuth, Ana C Stephens, Nicole M McNeil, and Martha W Alibali. Does understanding the equal sign matter? evidence from solving equations. *Journal for research in Mathematics Education*, 37(4):297–312, 2006. 2

[22] Elida V Laski and Qingyi Yu. Number line estimation and mental addition: Examining the potential roles of language and education. *Journal of experimental child psychology*, 117:29–44, 2014. 2

[23] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019. 1

[24] Kamil Mamak. Whether to save a robot or a human: on the ethical and legal limits of protections for robots. *Frontiers in Robotics and AI*, 8, 2021. 1

[25] Hiroaki Mikami, Kenji Fukumizu, Shogo Murai, Shuji Suzuki, Yuta Kikuchi, Taiji Suzuki, Shin-ichi Maeda, and Kohei Hayashi. A scaling law for synthetic-to-real transfer: How much is your pre-training effective? *arXiv preprint arXiv:2108.11018*, 2021. 3, 6

[26] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019. 3

[27] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, and 70 others. Scaling language models: Methods, analysis & insights from training gopher. DeepMind Research, 2021. 3, 6

[28] Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*, 2022. 1, 4

[29] Joelle Renstrom. Should robots have rights? *the Daily Beast*, 2018. 1

[30] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021. 1

[31] Barbara W Sarnecka and Susan Carey. How counting represents number: What children must learn and when they learn it. *Cognition*, 108(3):662–674, 2008. 2

[32] David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models, 2019. 2

[33] Utkarsh Sharma and Jared Kaplan. A neural scaling law from the dimension of the data manifold. *arXiv preprint arXiv:2004.10802*, 2020. 3

[34] AJ Sherman and S Seyfarth. Now is the time to figure out the ethical rights of robots in the workplace. *CNBC Op Ed*, 2018. 1

[35] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*, 2019. 5

[36] Alan M. Turing. Computing machinery and intelligence. *Mind*, LIX(236):433–460, 10 1950. 1

[37] Carissa Véliz. Moral zombies: why algorithms are not moral agents. *AI & SOCIETY*, 36(2):487–497, 2021. 1

[38] Cunxiang Wang, Boyuan Zheng, Yuchen Niu, and Yue Zhang. Exploring generalization ability of pretrained language models on arithmetic and logical reasoning. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 758–769. Springer, 2021. 1

[39] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 1

[40] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. 3