Impact of Pretraining Term Frequencies on Few-Shot Reasoning

Yasaman Razeghi [1] **Robert L. Logan IV** [1] **Matt Gardner** [2] **Sameer Singh** [1 3]

Abstract

Pretrained Language Models (LMs) have demonstrated ability to perform numerical reasoning by extrapolating from a few examples in few-shot settings. However, the extent to which this extrapolation relies on robust reasoning is unclear. In this paper, we investigate how well these models reason with terms that are less frequent in the pretraining data. In particular, we examine the correlations between the model performance on test instances and the frequency of terms from those instances in the pretraining data. We measure the strength of this correlation for a number of GPT-based language models (pretrained on the Pile dataset) on various numerical deduction tasks (e.g., arithmetic and unit conversion). Our results consistently demonstrate that models are more accurate on instances whose terms are more prevalent, in some cases above 70% (absolute) more accurate on the top 10% frequent terms in comparison to the bottom 10%. Overall, although LMs exhibit strong performance at few-shot numerical reasoning tasks, our results raise the question of how much models actually generalize beyond pretraining data, and we encourage researchers to take the pretraining data into account when interpreting evaluation results.

1. Introduction

Large language models have demonstrated outstanding performance in zero- and few-shot learning settings on numerous tasks, from simple classifications such as sentiment analysis to complex reasoning-related task like natural language inference and arithmetic (Brown et al., 2020; Radford et al., 2019). These results suggest that models may have gained the ability to perform simple inductive reasoning through a combination of pretraining and model size.

Q: What is 24 times 18? A: ___ *Model:* 432 ✓
Q: What is 23 times 18? A: ___ *Model:* 462 ✗

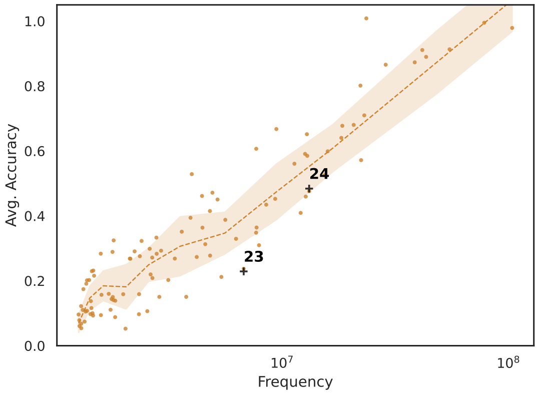


Figure 1. **Multiplication Performance:** Plot of GPT-J-6B's 2-shot accuracy on multiplication (averaged over multiple multiplicands and training instances) against the frequency of the equation's first term in the pretraining corpus. Each point represents the average performance for that term (e.g., 24) multiplied by numbers 1-50 and 5 choices of random seeds. As in the example, the performance difference for the numbers 24 and 23 is more than 20%. We find a strong correlation between accuracy and frequency.

However, current evaluation schemes for the reasoning of large language models, often neglect or underestimate the impact of data leakage from pretraining data when assessing their reasoning ability. Although the overlap between the training and evaluation splits of public datasets and their effect on the generalization of the language models have been studied (Elangovan et al., 2021; Lewis et al., 2020), the effect of the pretraining data has gotten less attention. Traditionally, a model that has *learned* to reason in the training phase should be able to generalize outside of the narrow context that it was trained in. Specifically, if the model has learned to reason, its performance on instances with less frequent terms (based on pretraining data) should not be significantly lower than its performance on the instances with more common terms.

As an illustration, consider the arithmetic task of multiplying two integers (shown in Figure 1). A model that has learned proper arithmetic skills should be able to answer the queries irrespective of the frequencies of the operands in the pretraining data. Therefore, it should have roughly equiv-

[1] Department of Computer Science, University of California, Irvine, USA [2] Microsoft Semantic Machines, USA [3] Allen Institute for Artificial Intelligence, USA. Correspondence to: Yasaman Razeghi <yrazeghi@uci.edu>.

alent performance when answering the queries *Q: what is* 24 *times* X*?* and *Q: what is* 23 *times* X*?*, when aggregated over various values of X. This is not the case with current LMs and we will study the effect of frequency terms in details through this paper. To show the effect of frequency, in this example, we plot the average accuracy of GPT-J-6B (Wang, 2021) on the numbers 0–100 (averaged over 1–50 as the other operand) against the frequency of the number in the pretraining data in Figure 1. We find a strong correlation between the term frequency and the model performance indicating that the model reasoning is not robust to these frequencies. Note that even "rare" terms are overall frequent (on the order of millions) in the pretraining data.

In this work, we investigate this impact of the frequency of test instance terms in the model's pretraining data on model's performance. We focus our analysis on numerical reasoning tasks, including addition, multiplication, and unit conversion. For each of these tasks, we identify relevant *terms* from each instance; for these tasks, terms are the numbers and units involved. We count occurrences of these terms in the pretraining data, including co-occurrences of term pairs or triples within a fixed window. This procedure allows us to aggregate over instances in which these terms appear and observe the relationship between term frequency and model accuracy on instances that include those terms. We summarize this behavior through the *performance gap* between instances that have the most frequent terms and instances that have the least frequent terms. Intuitively, models that exhibit a *high* performance gap are *more* accurate on instances that are more common in the pretraining data; this indicates that the model does not generalize appropriately and is likely affected by dataset overlap.

We present analysis on these numerical reasoning tasks for three sizes of the EleutherAI/GPT models pretrained on the Pile (Gao et al., 2020) dataset, which has been publicly released and thus permits this kind of analysis (in contrast to the data that, e.g., GPT-3 (Brown et al., 2020) was trained on). Our results show a consistently large performance gap between highest-frequency terms and lowest-frequency terms in all of our experiments; in some cases there is more than 70% of average accuracy gap between the first and last 10% indicating that even simple unigram statistics are highly correlated with models performance which should not happen if the model is performing reasoning. These observations suggest that any evaluation of reasoning that does not take the pretraining data into account is difficult to interpret, and that we need to revisit evaluation of language models with respect to their pretraining data.

2. Background and Methodology

Reasoning ability has long been considered as a proxy for intelligence (Johnson-Laird, 2010). Thus, developing models with this skill has been also an essential goal of AI and natural language processing (NLP) (Bommasani et al., 2021). Recently, large language models have exhibited an ability to perform reasoning-related tasks in few-shot settings without requiring any modifications to their parameters through a method called in-context learning. Our goal is to evaluate this reasoning skill in-depth for numerical induction tasks. This section provides background information on in-context learning and introduces our method for measuring the performance gap of the models on numerical reasoning tasks based on differences in pretraining term frequency.

2.1. In-context Learning

Brown et al. (2020) show that the large GPT-3 model is able to perform well on few-shot reasoning tasks without requiring any changes to its internal parameters, through the usage of a technique called *in-context learning*. In place of a typical learning procedure, in-context learning instead places training examples in a prompt format, which is subsequently fed to a language model as its input.

Among numerous experiments, Brown et al. (2020) show that GPT3 performs well on a variety of arithmetic questions such as addition and subtraction with 2–5 digit numbers. For example, they show that their largest model can perform zero-shot 2-digit addition with 76.9% accuracy. Although impressive, due to the large volume of data GPT-3 is trained on, it is possible that the model is merely repeating answers seen during pretraining. To attribute this performance to the model's *reasoning* capabilities, we need to make sure that the model is not affected by statistical overlaps between the terms of the arithmetic questions and the pretraining data.

In the following sections, we introduce metrics that we use to investigate the relationship between the frequency of terms in the pretraining data and the model performance on reasoning instances containing those terms. To assess this relation, we first define an approach for measuring term frequencies in a large pretraining dataset (Section 2.2). We connect these frequencies to reasoning performance by introducing the *performance gap* Δ (Section 2.3).

2.2. Frequency

We consider numerical reasoning tasks (Table 1) whose instances consist of input terms, $\mathbf{x} = (x_1, \ldots, x_i, \ldots x_n)$, and a derived output term y, where the x_i's are either positive integers or units of time (e.g., 1, 2, `hour`, etc.) and y is a positive integer. For example, for the task of multiplication, an instance might be $\mathbf{x} = (23, 18)$ and $y = 414$, representing the equation $23 \times 18 = 414$.

For each instance, we extract counts of the number of times that a subset of its terms $X \subseteq \{x_1, \ldots, x_n, y\}$ appear within a specified window in the pretraining data. We refer to this count as the frequency, ω_X, of X.

In this paper, we restrict our attention to frequencies involving three or less input terms, e.g., $\mathbf{x} = (x_1)$ or (x_1, x_2) or (x_1, x_2, x_3) and optionally the output term y, e.g.:

- $\omega_{\{x_1\}}$: the number of times that x_1 (e.g., 23) appears in the pretraining data.
- $\omega_{\{x_1,x_2\}}$: the number of times that the input terms x_1 (e.g., 23) and x_2 (e.g., 18) appear in the pretraining data within a specific window size.
- $\omega_{\{x_1,y\}}$: the number of times that the first input term x_1 (e.g., 23) and the output term y (e.g., 414) appear in the pretraining data within a specific window size.

Note that our usage of set notation in the subscript is deliberate; although $\mathbf{x} = (x_1, x_2)$ and $\mathbf{x}' = (x_2, x_1)$ are not necessarily the same (e.g., order is important when representing the task instance), frequency is symmetric (e.g., $\omega_{\{x_1,x_2\}} = \omega_{\{x_2,x_1\}} \forall x_1, x_2$).

2.3. Performance Gap

We want to measure how much *more* accurate the model is on instances containing more versus less frequent terms in the pretraining data. We do this by calculating the differences in average accuracies of the instances in the top and bottom quantiles of the distribution over term frequencies, which we call the *performance gap*.

Formally, let $\{(X^{(n)}, \omega_X^{(n)})\}$, $n \in [1, N]$, be a set of terms for a task and their associated term frequencies in the pretraining corpus. Given a task (e.g. addition), we create reasoning instances for each element of this set by instantiating uninstantiated values of x_i, and deriving y if $y \notin X^{(n)}$. We then measure the LM's accuracy $a^{(n)}$ over the set of instances, and repeat this process for all $n \in [1, N]$, producing a set $\Omega = \{(\omega_X^{(n)}, a^{(n)})\}$. The formula for the **performance gap** is then given by:

$$\Delta(\Omega) = \mathrm{Acc}(\Omega_{>90\%}) - \mathrm{Acc}(\Omega_{<10\%}) \qquad (1)$$

where $\Omega_{>90\%}$ is the top 10% of elements in Ω ordered by frequency, $\Omega_{<10\%}$ is the bottom 10%, and $\mathrm{Acc}(\Omega')$ is the average accuracy of elements in Ω'. We introduce the following convenient abuses of notation Δ_1, $\Delta_{1,2}$, $\Delta_{1,y}, \ldots$, to denote the performance gap over the frequency distributions of $\omega_{\{x_1\}}$, $\omega_{\{x_1,x_2\}}$, $\omega_{\{x_1,y\}}, \ldots$, respectively.

Concretely, for the multiplication example from Figure 1, $\mathbf{x} = (x_1, x_2)$ and we consider the performance gap over frequencies $\omega_{\{x_1\}}$. For each number (say 23), we count the number of times it appears in the pretraining corpus ($\omega_{\{23\}}$), and compute the average accuracy of the model over all instances where the first operand is 23. The performance gap w.r.t. to $\omega_{\{x_1\}}$ for this task is the difference between the average accuracy over the top 10% and the bottom 10% most frequent numbers in the pretraining corpus.

Table 1. Prompt templates and the number of test cases investigated for each numerical reasoning task.

Task	Prompt Template	#Test Cases
Arithematic		
Multiplication	*Q:What is x_1 times x_2? A: y*	5000
Addition	*Q:What is x_1 plus x_2? A: y*	5000
Operation Inference		
Mult. #	*Q:What is x_1 # x_2? A: y*	5000
Add. #	*Q:What is x_1 # x_2? A: y*	5000
Time Unit Inference		
Min→Sec	*Q:What is x_1 minutes in seconds? A: y*	79
Hour→Min	*Q:What is x_1 hours in minutes? A: y*	100
Day→Hour	*Q:What is x_1 days in hours? A: y*	100
Week→Day	*Q:What is x_1 weeks in days? A: y*	100
Month→Week	*Q:What is x_1 months in weeks? A: y*	100
Year→Month	*Q:What is x_1 years in months? A: y*	100
Decade→Year	*Q:What is x_1 decades in years? A: y*	100

3. Experiment Setup

In this section, we describe our setup to measure the effect of pretraining data on the few-shot evaluation of a number of numerical reasoning tasks for different language models. For reproducibility, we will release the complete source code of our experiments.

Language Models We experiment on models from EleutherAI i.e., GPT-J-6B (Wang, 2021), and GPT-Neo-1.3B, GPT-Neo-2.7B (Black et al., 2021). These models are publicly available, but more importantly, the corpus used to pretrain them has also been released. We use the HuggingFace[1] Transformer integration of the models in our experiments.

Pretraining Corpus Language models studied in this work are trained on the Pile dataset (Gao et al., 2020), a large-scale language modeling dataset (800GB) consisting of English documents in 22 academic or other professional data sources. Since our reasoning tasks focus on numbers, we count the frequency of all integers with less than seven digits using a white-space tokenizer. To calculate the frequencies of the numbers we use Amazon Elastic Map Reduce (EMR) platform. We will release the code and statistics of the frequencies.

Numerical Reasoning Tasks We create three types of datasets that target the mathematical capabilities of language models since solving mathematical questions is a useful *reasoning* capability of the models (Brown et al., 2020).

- **Arithmetic, 2 tasks** As the first task, we consider simple arithmetic operations: addition $x_1 + x_2 \rightarrow y$ and multipli-

[1] Source code at `https://huggingface.co/EleutherAI`

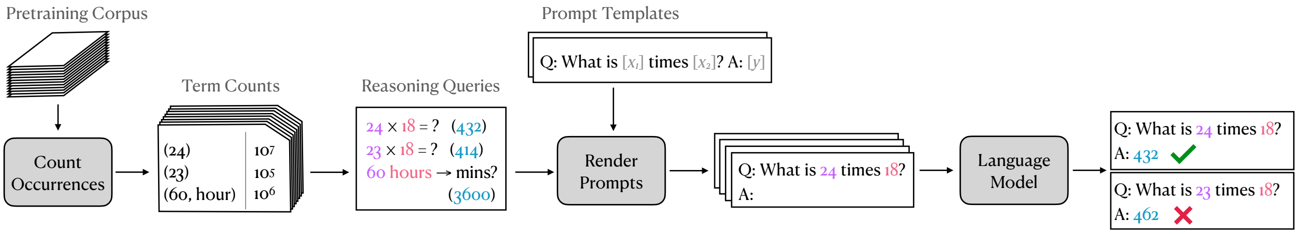


Figure 2. **Pipeline for Data Construction:** We use the term counts processed from the pretraining data to develop the reasoning queries and render them with prompts templates to a proper language model input format (illustrated using the example from Figure 1).

cation $x_1 \times x_2 \rightarrow y$. In both cases, the first operands (x_1) are numbers less than 100 (that are in the top 200 most frequent numbers) and the second operands (x_2) are the numbers in the range ($1 - 50$).

- **Operation Inference, 2 tasks** Instead of directly specifying the operation, we also create a variation where the model needs to *infer*, from a few examples, the operation itself, as well as the result, as introduced in the evaluation of Megatron-Turing model [2]. We replace the arithmetic operation with a "#", with the same operations and operands as previous, to create these datasets.
- **Time Unit Conversion, 7 tasks** Apart from directly executing arithmetic expressions, we are also interested in evaluating model capability to implicitly reason about these operations. To this end, we construct a unit conversion dataset by identifying the most frequent numbers that co-occur with time unit words ("second", "minute", "hour", "day", "week", "month", "year", and "decade") as the primary operand x_1, the time units themselves as additional operands ($x_2 \rightarrow x_3$), i.e. converting 24 hours to minutes is represented as (24, "hours", 60). We expect converting time values to be mathematically more straightforward than two-digit multiplication since the model need only multiply with the same (implicit) second operand, e.g., $\times 60$ for converting hours to minutes.

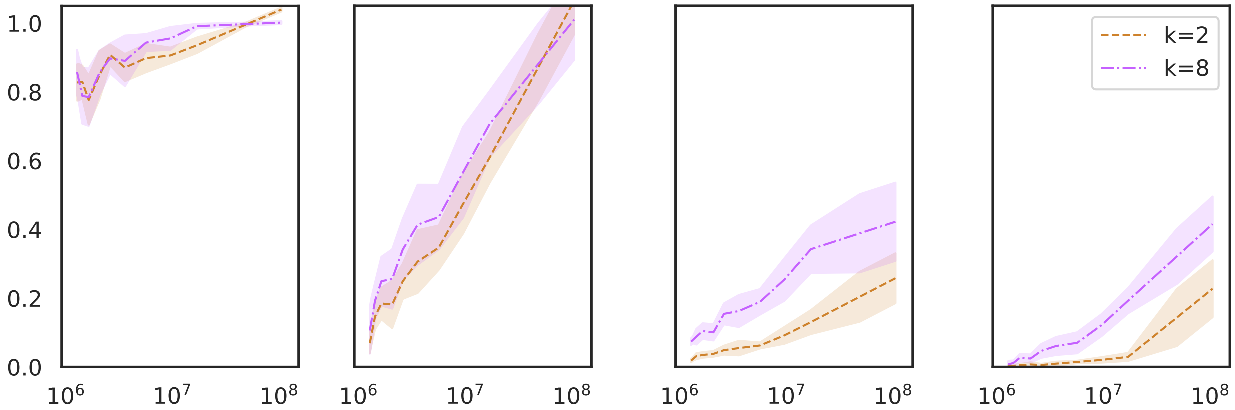
The overall pipeline for creating natural language instances for these datasets is illustrated in Figure 2. We compute occurrences and co-occurrences (if they are within a window of 5 words) of the terms in the corpus, i.e. the time units and numbers. We then generate instances for each of our reasoning datasets using the most frequent terms (in the top 200) as operands with less than 3 digits. We focus on the top terms since we expect the models to have a fairly reliable and robust representations for these words. Each reasoning instance is *rendered* as a natural language query using the prompt templates from Table 1, and fed to the language model to generate the answer. For example, to create a multiplication instance given the argument ($x_1 = 23, x_2 = 18$), we use the instance template to create a natural language input for the model as *"Q: What is* 23 *times* 18*? A:* __*"*, with the goal of producing "414" ($y = 23 \times 18 = 414$). For few-shot evaluation, we prompt the language models with $k = 0, 2, 4, 8, 16$ shots, and average performance over five random selection of the prompt instances, we randomly select k samples from the dataset as the training examples provided in the prompt and use the rest of the instances from the model as the test data.

4. Results

With the three types of reasoning tasks (consisting of 11 total datasets), we present an evaluation of the effect of pretraining frequency on the performance of the language models. For each dataset, we measure the performance gap on instances that consist of rarer (relatively) terms, for a few different choices of what to compute frequency over i.e different combinations of the instance terms. To observe this effect with larger language models, we also investigate the effect of the model size on this performance gap.

Arithmetic We first study the performance on simple addition and multiplication of numbers. The results for the GPT-J-6B model is provided in Table 2, with performance gap computed just for x_1, for (x_1, x_2), and for (x_1, y). In *multiplication*, we observe a very high performance gap for all these definitions of frequencies, suggesting a strong effect of frequency in the pretraining data on the model's ability to perform multiplication. For better illustration of the performance gap, we plot the mean accuracy across the frequency of x_1 in Figure 3b. The plot demonstrates the strong correlation between the models accuracy on specific instances, and the instance element frequency in the pretraining data. For *addition*, we observe an overall higher performance of the GPT-J-6B model in comparison to the multiplication experiments. However, the performance gap on all of the definitions of the instance frequencies still shows an strong effect on the models accuracy. As shown in Figure 3a, the average accuracy of the model still has a positive slope, indicating the effect of instance frequencies.

[2] https://turing.microsoft.com/



(a) Arithmetic-Addition (b) Arithmetic-Multiplication (c) Op.Inference-Addition (d) Op. Inference-Multiplication

Figure 3. **The GPT-J-6B accuracy on arithematic and operator inference** tasks, with k shots. The average accuracy (y-axis) of the binned instances is highly correlated with their term frequencies $\omega_{\{x_1\}}$ in the pretraining corpus (x-axis).

Table 2. **GPT-J-6B results on arithmetic, operation inference (#) tasks** Δ_1, $\Delta_{1,2}$ and $\Delta_{1,y}$ represent the performance gap over the frequency distributions of $\omega_{\{x_1\}}$, $\omega_{\{x_1,x_2\}}$ and $\omega_{\{x_1,y\}}$ respectively. x_1 represent the first operand, x_2 second operand and y the answer of the arithmetic question.

k	Multiplication				Addition				Multiplication (#)				Addition (#)			
	Acc.	Δ_1	$\Delta_{1,2}$	$\Delta_{1,y}$	Acc.	Δ_1	$\Delta_{1,2}$	$\Delta_{1,y}$	Acc.	Δ_1	$\Delta_{1,2}$	$\Delta_{1,y}$	Acc.	Δ_1	$\Delta_{1,2}$	$\Delta_{1,y}$
0	5.4	18.0	20.6	30.8	1.6	8.4	6.9	8.0	-	-	-	-	-	-	-	-
2	35.9	77.6	79.3	89.9	88.2	16.8	21.7	21.9	3.1	14.1	13.7	14.2	7.8	18.1	25.3	28.3
4	39.2	70.8	76.4	83.5	91.4	15.0	24.8	26.4	5.7	20.9	21.3	23.4	9.8	24.8	30.1	30.4
8	42.9	74.6	80.8	86.0	89.6	16.3	26.5	29.6	9.4	31.3	33.2	34.7	19.8	31.0	44.8	45.2
16	40.9	73.3	77.7	82.6	88.6	16.4	27.3	31.0	11.0	39.6	38.7	42.6	26.2	38.5	47.2	49.9

Operation Inference These tasks aim to assess the model capability to both infer the math operation and to perform the actual computation. Overall, as we see in Table 2, the model is much less accurate here as compared to the arithmetic experiments. However, the model has better performance on the frequent instances even for these low performance tasks (see detailed trend in Figures 3d and 3c). The performance gap here suggests that the effect of pretraining is not only for tasks that the model is accurate on, but even for operation inference that is more challenging and require deeper reasoning. An additional interesting trend is in the last column of Table 2 with the steady growth of performance gap with respect to $\omega_{\{x_1,y\}}$ (co-occurrence of the instance input element and the true answer) as the number of shots increases and the model gets better in performance. Moreover, the lower accuracy here as compared to addition experiments in the previous section suggests that the model is unable to infer the operation from the few-shot prompts, and it may be performing some form of pattern matching based on the pretraining data.

Time-Unit Conversion The performance gap evaluated on all the time unit conversion experiments is provided in Table 3. We first observe a relatively high performance gap on all the tasks except the conversion from decade to year. For example, Figure 4 illustrates the trend of increased model performance with the frequency of instance elements for converting the time values from years to months. We also observe a general pattern of increase in the performance gap as the number of shots (training examples in the prompt) increases (results are in table 3). These results suggest that even though the model gets more accurate, the improvements focus on more frequent instances of the task. *Decades to years:* As we can observe in Figure 5, The model performs nearly perfectly on this task with as few as 8 shots, and we only see very small performance gap s. This is likely due to the task being quite simple (appending a “0” to the input number) and the model is able to *generalize* in the manner we are evaluating it. However, it is also possible that we are simply not identifying the right frequency statistics for this task, and there is an effect that our current evaluation setup does not capture.

Studying the Size of Language Models To further study the impact of language models sizes on the performance gap caused by the instance frequencies, we perform the

Table 3. **GPT-J-6B results on Time-Unit Conversion:** $\Delta_{1,2}$, $\Delta_{1,2,3}$ and $\Delta_{1,2,y}$ represent the performance gap over the frequency distributions of $\omega_{\{x_1,x_2\}}$, $\omega_{\{x_1,x_2,x_3\}}$ and $\omega_{\{x_1,x_2,y\}}$ respectively, where x_1 is the number operand, x_2 is the source time unit, x_3 is the second implicit number operand needed for performing the conversion and the y is the true answer.

k	Min→Sec				Hour→Min				Day→Hour				Week→Day			
	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$
0	1.3	0.0	0.0	12.5	1.0	0.0	0.0	5.0	1.0	0.0	0.0	10.0	1.0	0.0	0.0	10.0
2	25.5	62.5	67.5	67.5	19.4	58.0	40.5	44.0	12.1	28.9	24.0	28.0	13.1	43.5	50.0	54.0
4	35.5	60.0	71.7	63.1	29.1	76.4	50.5	59.0	22.7	46.4	45.0	47.5	19.2	40.9	43.3	47.0
8	49.9	72.1	79.0	52.7	36.3	74.6	52.5	63.0	31.0	59.1	52.5	54.5	28.6	70.6	62.0	67.0
16	58.4	82.7	74.4	48.5	42.8	80.1	49.0	62.5	43.3	62.8	56.0	54.8	28.0	22.1	31.4	33.2

Shots, k	Month→Week				Year→Month				Decade→Year			
	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$
0	1.0	0.0	0.0	10.0	1.0	0.0	0.0	10.0	3.1	14.3	14.3	28.6
2	30.1	8.5	9.3	21.0	21.8	58.0	64.0	53.0	76.5	38.8	47.1	43.1
4	63.3	22.9	26.2	10.5	31.9	64.8	69.5	66.8	96.7	2.9	0.0	2.9
8	80.9	33.8	30.8	24.0	45.4	55.0	72.0	50.0	99.6	0.0	0.0	0.0
16	84.5	43.4	57.0	30.3	56.7	58.7	65.3	61.3	100.0	0.0	0.0	0.0

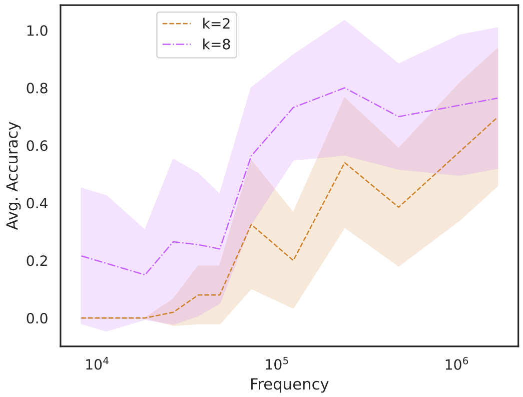


Figure 4. **GPT-J-6B performance on Year→Month:** The interpolation lines show the correlation between the average accuracy and the $\omega_{\{x_1,x_2\}}$. k is the number of shots.

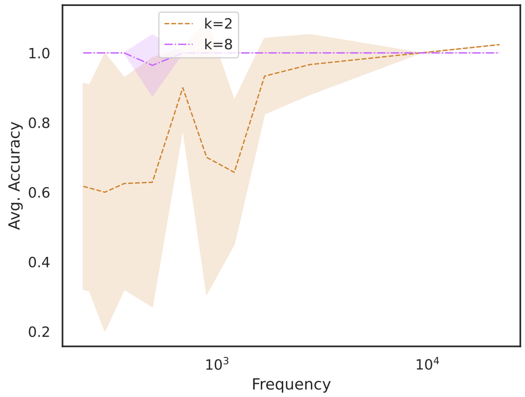
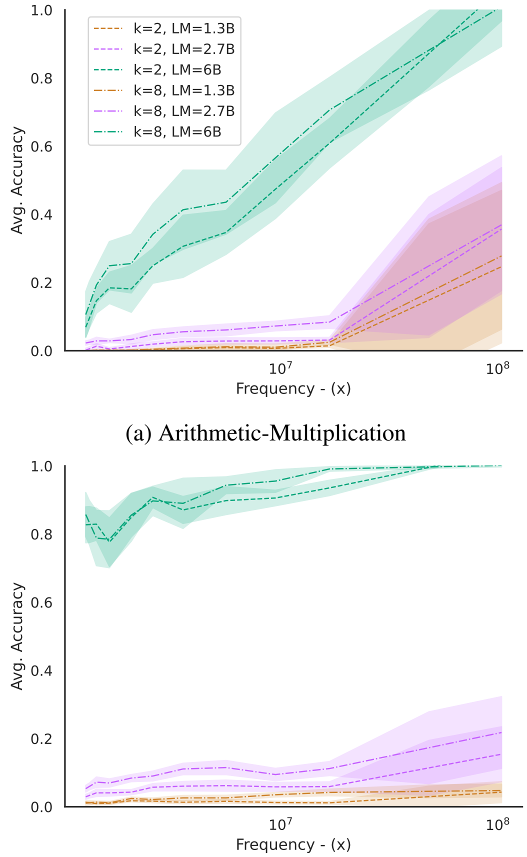


Figure 5. **GPT-J-6B performance on Decade→Year:** The interpolation average accuracy line over the $\omega_{\{x_1,x_2\}}$ show that the model reaches a high performance with the number of shots $k = 8$, there is still a performance gap in the case of $k = 2$.

arithmetic experiments for 2 and 8 shots using a variety of models (including the smaller versions of models GPT-Neo-1.3B and GPT-Neo-2.7B). We can see the trends of the average accuracy of the models in Figures 6a and 6b. The smaller models overall are less accurate on the arithmetic tasks, which is consistent with observations in related work (Brown et al., 2020). However, their success is still focused on the more frequent terms from the pretraining corpus, suggesting that even the smaller models show the effect of reliance on the pretraining data, although to a much lower extent than the larger ones.

Summary Overall, we observe high positive performance gap for almost all of the experiments on the three definition levels of the frequency for each task. This suggests a strong effect of frequency of the instances in the pretraining data on the model performance. In particular, evaluation using performance gap with $\omega_{\{x_1\}}$ shows that even the *unigram* statistics of the instances have strong correlation with the models performance on the instance.

Other than some exceptional cases, we observe an increasing trend in the performance gap as we put more training instances in the prompt (the number of shots); this can be a further indication that the model is directed through the patterns in the pretraining data to answer the reasoning questions. Our experiments with the smaller sizes of the model also show that they can only solve the frequent instances of the tasks, which further supports our observation that model performance is correlated with the term frequencies.



(a) Arithmetic-Multiplication

(b) Arithmetic-Addition

Figure 6. **The effect of model size on performance:** Smaller models only perform well on instances with more frequent terms in the pretraining data. k represents the number of shots.

5. Related Work

A large and growing body of literature has investigated a number of related concerns with large language models.

Prompting Prompting has been widely applied to study the factual (Petroni et al., 2019), commonsense (Davison et al., 2019; Weir et al., 2020; Lin et al., 2020), mathematical (Saxton et al., 2019), and other NLP task-related (Radford et al., 2019; Shin et al., 2020) knowledge LMs acquire during pretraining. In this work, we focus on the *in-context learning* setup of Brown et al. (2020), who use prompts that include training examples to diagnose LMs' few-shot learning capabilities.

Impact of Frequency on LM Performance Kassner et al. (2020) and Wei et al. (2021) perform controlled experiments varying pretraining data to characterize the extent pretraining affects LMs' ability to learn to memorize and reason with facts as well as learn generalizable syntax rules. In line with our results, both of these works find that frequency is a distinguishing factor in whether or not the model memorizes a particular fact or syntactic rule for a verb form. Sinha et al. (2021) further demonstrate that shuffling word order during pretraining has minimal impact on an LMs' accuracy on downstream tasks, and, concurrent with this work, Min et al. (2022) similarly find that shuffling labels in in-context learning demonstrations has a minimal impact on few-shot accuracy. These results further suggest that LMs' performance is largely driven by their ability to model high-order word co-occurrence statistics. Although frequent terms are more likely to be memorized, data privacy researchers have also shown that LMs may memorize sensitive sequences occurring in training data (e.g., social security and credit card numbers), even if they are rare (Carlini et al., 2019; Song & Shmatikov, 2019).

Memorization Feldman (2020) provide a theoretical definition of memorization as the difference between the accuracy of a model on a training data point when that point is included vs. excluded from training. In subsequent work, they develop an approach for approximating memorization using influence functions Feldman & Zhang (2020). This framework is applied to study memorization in language models by Zhang et al. (2021), who find that training examples that are memorized by the LM tend to have high influence of LM predictions on similar validation instances. Their result may provide a plausible explanation that the frequency effects observed in this work are due to memorization.

Training Artifacts Challenge Evaluation Our results raise the issue that in-context learning probes may overestimate an LM's ability generalize from few examples when biases are present in the training data. This is consistent with prior work that has exposed the similar effects of biases from: lexical cues in natural language inference datasets (Gururangan et al., 2018; Poliak et al., 2018; McCoy et al., 2019), question-passage overlap and entity cues in reading comprehension datasets (Chen et al., 2016; Sugawara et al., 2018; Jia & Liang, 2017; Lewis et al., 2021), gender cues in coreference resolution datasets (Rudinger et al., 2018), popularity in named entity disambiguation (Chen et al., 2021), similarity between training and test instances in information extraction and sentiment analysis datasets (Elangovan et al., 2021), and effects of how data is split (Gorman & Bedrick, 2019; Søgaard et al., 2021). Relatedly, data poisoning research studies how to adversarially introduce artifacts into training data to produce unwanted model behaviors (Nelson et al., 2008; Chan et al., 2020; Wallace et al., 2021). A general statistical procedure to test for artifacts is presented in Gardner et al. (2021), who also theoretically show that large datasets are almost certain to contain artifacts under reasonable assumptions. Techniques for mitigating biases in the presence of dataset artifacts are covered by Romanov et al. (2019) and Karimi Mahabadi et al. (2020).

Documenting Pretraining Data To better understand the risks of dataset artifacts, there has been a call to better document the characteristics and intended uses of datasets (Gebru et al., 2021; Bender et al., 2021). However, due to the sheer size of LM pretraining datasets—which range from 100's of GBs to 10's of TBs—doing so can pose a substantial challenge. Despite this, researchers have been able to estimate word frequencies, topics, and genres of documents (Sharoff, 2020), as well as proportions of toxic text (Gehman et al., 2020) appearing in OpenWebText (Gokaslan & Cohen, 2019). Similar efforts have been made to characterize the top-level domains, amount of hate speech, and censured text appearing in the C4 corpus (Raffel et al., 2020; Dodge et al., 2021; Luccioni & Viviano, 2021). Our work documents co-occurrence statistics of numbers and dates of documents appearing in the Pile dataset.

Numeracy and Temporal Reasoning in LMs Our work contributes a larger body of work dedicated to studying numeracy in word embeddings and language models (Spithourakis & Riedel, 2018; Wallace et al., 2019). Recently, Geva et al. (2020) and Zhou et al. (2020) have proposed training schemes to help improve LMs' temporal and numerical reasoning capabilities. Patel et al. (2021) also showed that NLP math solvers rely on simple heuristics to answer math questions. We expect that the performance gap metric proposed in this work will be useful to better understand the impact of such schemes.

6. Discussion and Future Work

In this work, we consider how to conduct few-shot evaluations in light of the analysis with respect to the pretraining data. Prior work has attempted to control for overlap between training or pretraining data and the test instances, but as we have seen, those methods are insufficient. For example, Brown et al. (2020) measure the impact of removing instances from evaluation datasets that share 13-gram overlap with their pretraining data on GPT-3's accuracy, and also argue that the low occurrence of exact phrases such as "NUM1 + NUM2 =" and "NUM1 plus NUM2" in the pretraining data indicate that the model's strong performance on arithmetic tasks is likely due to factors other than memorization. However, we have seen that model performance is impacted by much simpler statistical patterns, as small as unigram overlaps with the pretraining data.

For these reasons, we strongly recommend that evaluation of reasoning capabilities should take the pretraining corpus into account, and any claims of reasoning can only be made after demonstrating robustness to the effect of pretraining. LM benchmarks typically have no reference to the model's pretraining data. However, it is impossible to interpret few-shot performance on any benchmark without reference to information from the data that the LM was trained on. One possible addition to future evaluation is the performance gap between high-frequency and low-frequency terms, perhaps including only terms that have been seen more than some threshold value. It is worth mentioning that, even a performance gap of 0 is likely not sufficient to demonstrate a claim of reasoning capabilities—what exactly constitutes "reasoning" remains ill-defined—but it may be a necessary condition, and one that current models do not meet.

There are a few limitations to our study that open up avenues for future research. We are not making a *causal* claim here, and in general, there may be confounders that we have not eliminated in our study. We recommend further research in investigating methods in causal inference and interventions during training to provide finer-grained analysis of the effect of pretraining. Since our approach aggregates fairly simple patterns, the effect we observe might be stronger if a wider variety and complexity of patterns is considered in the pretraining corpus. Similarly, our work was also limited to numerical reasoning tasks, and it would be worthwhile to consider how much other reasoning capability evaluations are impacted by the same effect, which could be measured using the performance gap metric introduced here. Defining appropriate instance terms for other reasoning tasks such as commonsense reasoning will be a challenging but important direction for future work. With the insights in this work and these recommendations, we hope to inspire further studies into the effect of pretraining on LM performance.

7. Conclusion

We show that in-context language model performance on numerical reasoning tasks can be impacted significantly by low-order co-occurrence statistics in the pretraining data, raising questions on the extent to which these models are actually *reasoning* to solve these tasks. These observations suggest the necessity for reconsidering and redefining the reasoning evaluation schemes for the large language models. Further characterizing the impacting factors on the models reasoning capacities is also an important tasks for the community. Most importantly, we suggest that the NLP community should not treat the pretraining data of the large language models as unknown black boxes. Overlooking the impact of the pretraining data can be misleading in evaluating the model reasoning skills.

Acknowledgements

We would like to thank Yanai Elazar, Mukund Sundarajan, Marco Tulio Ribeiro, Eric Wallace, Shivanshu Gupta, Navid

Salehnamadi, Pouya Pezeshkpour, and Dylan Slack for valuable discussions and feedback on this work. This material is sponsored in part by the DARPA MCS program under Contract No. N660011924033 with the United States Office Of Naval Research, by an Amazon Research Award, and by awards IIS-2046873 and IIS-204098 from the National Science Foundation.

References

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623, 2021.

Black, S., Gao, L., Wang, P., Leahy, C., and Biderman, S. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL https://doi.org/10.5281/zenodo.5297715.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Carlini, N., Liu, C., Erlingsson, U., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, pp. 267–284, USA, 2019. USENIX Association. ISBN 9781939133069.

Chan, A., Tay, Y., Ong, Y.-S., and Zhang, A. Poison attacks against text datasets with conditional adversarially regularized autoencoder. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4175–4189, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.373. URL https://aclanthology.org/2020.findings-emnlp.373.

Chen, A., Gudipati, P., Longpre, S., Ling, X., and Singh, S. Evaluating entity disambiguation and the role of popularity in retrieval-based NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4472–4485, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.345. URL https://aclanthology.org/2021.acl-long.345.

Chen, D., Bolton, J., and Manning, C. D. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2358–2367, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1223. URL https://aclanthology.org/P16-1223.

Davison, J., Feldman, J., and Rush, A. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1173–1178, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1109. URL https://aclanthology.org/D19-1109.

Dodge, J., Sap, M., Marasović, A., Agnew, W., Ilharco, G., Groeneveld, D., Mitchell, M., and Gardner, M. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1286–1305, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.98. URL https://aclanthology.org/2021.emnlp-main.98.

Elangovan, A., He, J., and Verspoor, K. Memorization vs. generalization : Quantifying data leakage in NLP performance evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1325–1335, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.113. URL https://aclanthology.org/2021.eacl-main.113.

Feldman, V. *Does Learning Require Memorization? A Short Tale about a Long Tail*, pp. 954–959. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450369794. URL https://doi.org/10.1145/3357713.3384290.

Feldman, V. and Zhang, C. What neural networks memorize and why: Discovering the long tail via influence estimation. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2881–2891. Curran Associates, Inc., 2020. URL https://proceedings.

neurips.cc/paper/2020/file/1e14bfe2714193e7af5abc64ecbd6b46-Paper.pdf.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., and Leahy, C. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gardner, M., Merrill, W., Dodge, J., Peters, M., Ross, A., Singh, S., and Smith, N. A. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1801–1813, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.135. URL https://aclanthology.org/2021.emnlp-main.135.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., au2, H. D. I., and Crawford, K. Datasheets for datasets, 2021.

Gehman, S., Gururangan, S., Sap, M., Choi, Y., and Smith, N. A. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3356–3369, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.301. URL https://aclanthology.org/2020.findings-emnlp.301.

Geva, M., Gupta, A., and Berant, J. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 946–958, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.89. URL https://aclanthology.org/2020.acl-main.89.

Gokaslan, A. and Cohen, V. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Gorman, K. and Bedrick, S. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2786–2791, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1267. URL https://aclanthology.org/P19-1267.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://aclanthology.org/N18-2017.

Jia, R. and Liang, P. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL https://aclanthology.org/D17-1215.

Johnson-Laird, P. N. Mental models and human reasoning. *Proceedings of the National Academy of Sciences*, 107 (43):18243–18250, 2010. ISSN 0027-8424. doi: 10.1073/pnas.1012933107. URL https://www.pnas.org/content/107/43/18243.

Karimi Mahabadi, R., Belinkov, Y., and Henderson, J. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8706–8716, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.769. URL https://aclanthology.org/2020.acl-main.769.

Kassner, N., Krojer, B., and Schütze, H. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pp. 552–564, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.conll-1.45. URL https://aclanthology.org/2020.conll-1.45.

Lewis, P., Stenetorp, P., and Riedel, S. Question and answer test-train overlap in open-domain question answering datasets. *arXiv preprint arXiv:2008.02637*, 2020.

Lewis, P., Stenetorp, P., and Riedel, S. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1000–1008, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.86. URL https://aclanthology.org/2021.eacl-main.86.

Lin, B. Y., Lee, S., Khanna, R., and Ren, X. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6862–6868, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.

557. URL https://aclanthology.org/2020.emnlp-main.557.

Luccioni, A. and Viviano, J. What's in the box? an analysis of undesirable content in the Common Crawl corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 182–189, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.24. URL https://aclanthology.org/2021.acl-short.24.

McCoy, T., Pavlick, E., and Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://aclanthology.org/P19-1334.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Nelson, B., Barreno, M., Chi, F. J., Joseph, A. D., Rubinstein, B. I. P., Saini, U., Sutton, C., Tygar, J. D., and Xia, K. Exploiting machine learning to subvert your spam filter. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, LEET'08, USA, 2008. USENIX Association.

Patel, A., Bhattamishra, S., and Goyal, N. Are nlp models really able to solve simple math word problems?, 2021.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Van Durme, B. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pp. 180–191, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-2023. URL https://aclanthology.org/S18-2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

Romanov, A., De-Arteaga, M., Wallach, H., Chayes, J., Borgs, C., Chouldechova, A., Geyik, S., Kenthapadi, K., Rumshisky, A., and Kalai, A. What's in a name? Reducing bias in bios without access to protected attributes. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4187–4195, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1424. URL https://aclanthology.org/N19-1424.

Rudinger, R., Naradowsky, J., Leonard, B., and Van Durme, B. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 8–14, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2002. URL https://aclanthology.org/N18-2002.

Saxton, D., Grefenstette, E., Hill, F., and Kohli, P. Analysing mathematical reasoning abilities of neural models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=H1gR5iR5FX.

Sharoff, S. Know thy corpus! robust methods for digital curation of web corpora. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 2453–2460, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.298.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., and Singh, S. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4222–4235, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.346. URL https://aclanthology.org/2020.emnlp-main.346.

Sinha, K., Jia, R., Hupkes, D., Pineau, J., Williams, A., and Kiela, D. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 2888–2913, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.230. URL https://aclanthology.org/2021.emnlp-main.230.

Søgaard, A., Ebert, S., Bastings, J., and Filippova, K. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1823–1832, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.156. URL https://aclanthology.org/2021.eacl-main.156.

Song, C. and Shmatikov, V. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 196–206, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330885. URL https://doi.org/10.1145/3292500.3330885.

Spithourakis, G. and Riedel, S. Numeracy for language models: Evaluating and improving their ability to predict numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2104–2115, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1196. URL https://aclanthology.org/P18-1196.

Sugawara, S., Inui, K., Sekine, S., and Aizawa, A. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4208–4219, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1453. URL https://aclanthology.org/D18-1453.

Wallace, E., Wang, Y., Li, S., Singh, S., and Gardner, M. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5307–5315, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1534. URL https://aclanthology.org/D19-1534.

Wallace, E., Zhao, T., Feng, S., and Singh, S. Concealed data poisoning attacks on NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 139–150, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.13. URL https://aclanthology.org/2021.naacl-main.13.

Wang, B. Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

Wei, J., Garrette, D., Linzen, T., and Pavlick, E. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 932–948, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.72. URL https://aclanthology.org/2021.emnlp-main.72.

Weir, N., Poliak, A., and Durme, B. V. Probing neural language models for human tacit assumptions. In *CogSci*, 2020.

Zhang, C., Ippolito, D., Lee, K., Jagielski, M., Tramèr, F., and Carlini, N. Counterfactual memorization in neural language models. *arXiv preprint arXiv:2112.12938*, 2021.

Zhou, B., Ning, Q., Khashabi, D., and Roth, D. Temporal common sense acquisition with minimal supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7579–7589, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.678. URL https://aclanthology.org/2020.acl-main.678.