An Ill-Designed Study of Math Word Problems in Large Language Models: Review of (Ye, Xu, Li, and Allen-Zhu, 2024)

Ernest Davis
Dept. of Computer Science
New York University
New York, NY 10012
davise@cs.nyu.edu

August 3, 2024

Abstract

The article "Physics of Language Models: Part 2.1, Grade-School Math and the Hidden Reasoning Process" (Ye, Xu, Li, and Allen-Zhu, 2024) describes a number of experiments in which a transformer architecture was trained from scratch and tested on a collection of synthesized elementary math problems. Based on the results of these experiments, they make a number of large claims about the capacity of large language models for solving elementary math problems.

However, the training and test sets are extremely unnatural in several respects, and entirely different from actual mathematical word problems. It is also, despite the authors' claims, enormously less diverse than any of the standard benchmark collections of math word problems. Since arithmetic is carried out modulo 23, the arithmetic operations are entirely ungrounded any real world situation. Moreover the model being trained, though a transformer, is not a large language model. I argue, therefore, that most of the results claimed are untrue or misleading, and that these experiments are almost entirely irrelevant to an understanding of the capacities of large language models on elementary math problems.

1 Introduction

The capacity of large language models (LLMs), such as GPT-k, to solve word problems at the grade-school level has been extensively studied (e.g. Mishra, 2022). As of August 2024, state-of-the-art LLMs achieve a high level of performance over standard benchmarks for these. However, it is hard to determine the significance of this success for a number of reasons.

- There is evidence (Zhang et al. 2024) that some of the large math problem datasets such as GSM8K (Cobbe et al. 2021) have been included in the training set for leading LLMs and that this data contamination is responsible for some of the high levels of performance.
- The most powerful and best known LLMs, such as GPT-4, are not available to researchers for any kind of experiment, other than off-the-shelf execution, and would in any case be too large for controlled experiments involving training from scratch.
- Some of the best known math problem datasets are flawed. For example, some have wording artifacts that can give an AI a clue as to which operations are required (Patel, Bhattamishra, and Goyal, 2023).

In order to be able to study the workings of an LLM on math problems in depths and to avoid the problem of data contamination, Ye, Xu, Li, and Allen-Zhu (2024) developed a system for synthesizing word problems, which they used to train and test a modified GPT-2 transformer model from scratch. Extending the work of (Hewitt and Manning, 2019), they developed a new method for probing what features of the problem are being computed in which parts of the neural network. Based on the design and results of their experiment, they make a number of strong claims:

1. Their dataset is "highly diverse".
2. The trained model achieves both 99% accuracy on in-distribution test examples and high accuracy on out-of-distribution test examples.
3. "[T]he model can learn to generate shortest solutions, almost always avoiding unnecessary computations."
4. "[T]he model (mentally!) preprocesses the full set of necessary parameters before it starts any generation."
5. "Before any question is asked, it already (mentally!) computes with good accuracy which parameters depend on which, even though some are not needed for solving the math problem."
6. "We explain why mistakes occur."
7. "The depth of the language model is crucial for its reasoning ability."

However, on examination, their experiment is seriously flawed, and the claimed results are, with a single exception (result 3) either false, misleading, or unimportant. The chief problem is that the problems in their dataset are extremely bizarre and bear no resemblance to actual mathematical word problems. For that reason, the experiment cannot possibly shed any light on the capacity of any AI system to deal with natural mathematical word problems. All in all, I have never seen another paper on math word problems that so entirely ignored key realities of math word problems; nor another paper on any aspect of language that so entirely ignored key realities of human language.

Result 3 above is, indeed, correctly formulated, supported by the experimental results, and moderately interesting. However, none of the other claims hold water. Result 1 is not only false but the reverse of the truth; the dataset is incomparably less diverse and more uniform than any of the standard datasets of math word problems. In result 2, the accuracy is unsurprising and the sense in which the test set was "out of distribution" is entirely unimportant; the problems in the test set in fact only differ from those in the training set by the number of sentences, not by the type of sentences. Result 4 is a consequence of result 3 and adds nothing further. Result 5 rests on a misconstrual of the task involved in training. The AI is not being trained to solve the math problem; it is being trained to predict the next token, and probably the parameter dependencies are relevant to next-token prediction. Result 6 is very much overstated. Result 7 is later qualified to a degree that makes it irrelevant to the workings of actual large language models. I will discuss these in more detail in section 4

Furthermore, the architecture that they have used is a transformer model but it is not a large language model; it is not a language model at all. Consequently, it is very doubtful to what extent results obtained from the experiments apply to large language models, except that, if it is shown that the system here *does* do something, that means that an LLM *might* be able to do it.

2 The Problems in the Dataset

The paper contains two examples of the problems in the dataset that they used, together with their solutions:

Problem: An Easy Example The number of each Riverview High's Film Studio equals 5 times as much as the sum of each Film Studio's Backpack and each Dance Studio's School Daypack. The number of each Film Studio's School Daypack equals 12 more than the sum of each Film Studio's Messenger Backpack and each Central High's Film Studio. The number of each Central High's Film Studio equals the sum of each Dance Studio's School Daypack and each Film Studio's Messenger Backpack. The number of each Riverview High's Dance Studio equals the sum of each Film Studio's Backpack, each Film Studio's Messenger Backpack, each Film Studio's School Daypack and each Central High's Backpack. The number of each Dance Studio's School Daypack equals 17. The number of each Film Studio's Messenger Backpack equals 13. How many Backpack does Central High have?

Solution: An Easy Example Define Dance Studio's School Daypack as p; so p = 17. Define Film Studio's Messenger Backpack as W; so W = 13. Define Central High's Film Studio as B; so B = p + W = 17 + 13 = 7. Define Film Studio's School Daypack as g; R = W + B = 13 + 7 = 20; so g = 12 + R = 12 + 20 = 9. Define Film Studio's Backpack as w; so w = g + W = 9 + 13 = 22. Define Central High's Backpack as c; so c = B * w = 7 * 22 = 16. Answer: 16.

Problem: A Hard Example The number of each Jungle Jim's International Market's Cheese equals the sum of each Parmesan Cheese's Pear and each The Fresh Market's Ice Cream. The number of each Ice Cream's Pineapple equals 2 more than each Goat Cheese's Grape. The number of each New Seasons Market's Goat Cheese equals the sum of each Residential College District's Jungle Jim's International Market, each Jungle Jim's International Market's Parmesan Cheese and each Residential College District's Supermarket. The number of each Arts Campus's New Seasons Market equals each Cheese's Pineapple. The number of each Goat Cheese's Banana equals each Vocational School District's Product. The number of each Residential College District's Jungle Jim's International Market equals 5 more than each Ice Cream's Grape. The number of each Parmesan Cheese's Pineapple equals each Parmesan Cheese's Pear. The number of each Residential College District's The Fresh Market equals each Arts Campus's Trader Joe's. The number of each Arts Campus's Trader Joe's equals each Parmesan Cheese's Ingredient. The number of each Goat Cheese's Grape equals 0. The number of each The Fresh Market's Ice Cream equals 13 more than the difference of each Residential College District's The Fresh Market and each Parmesan Cheese's Grape. The number of each Goat Cheese's Pineapple equals each New Seasons Market's Product. The number of each Vocational School District's The Fresh Market equals the sum of each Trader Joe's's Cheese and each The Fresh Market's Cheese. The number of each Trader Joe's's Cheese equals 6. The number of each The Fresh Market's Cheese equals 3. The number of each Jungle Jim's International Market's Ice Cream equals the difference of each Ice Cream's Banana and each Goat Cheese's Grape. The number of each Jungle Jim's International Market's Parmesan Cheese equals each Ice Cream's Pineapple. The number of each Parmesan Cheese's Pear equals the difference of each Goat Cheese's Grape and each Ice Cream's Grape. The number of each Parmesan Cheese's Grape equals 12 times as much as each Residential College District's Jungle Jim's International Market. The number of each The Fresh Market's Parmesan Cheese equals each The Fresh Market's Cheese. The number of each Ice Cream's Banana equals the sum of each Parmesan Cheese's Pineapple and each Ice Cream's Pineapple. The number of each School District's Jungle Jim's International Market equals each The Fresh Market's Ice Cream. The number of each Cheese's Pineapple equals 20 more than the sum of each Trader Joe's's Cheese and each The Fresh Market's Cheese. The number of each Trader Joe's's Parmesan Cheese equals 16. The number of each Ice Cream's Pear equals 8. The number of each Ice Cream's Grape equals each Goat Cheese's Grape. How many Product does School District have?

Solution A Hard Example Define Goat Cheese's Grape as u; so u = 0. Define Ice Cream's Grape as x; so x = u = 0. Define Residential College District's Jungle Jim's International Market as N; so N = 5 + x = 5 + 0 = 5. Define Parmesan Cheese's Pear as G; so G = u- x = 0 - 0 = 0. Define Parmesan Cheese's Grape as f; so f = 12 * N = 12 * 5 = 14. Define Parmesan Cheese's Pineapple as C; so C = G = 0. Define Parmesan Cheese's Ingredient as Z; e = f + C = 14 + 0 = 14; so Z = e + G = 14 +

0 = 14. Define Arts Campus's Trader Joe's as q; so q = Z = 14. Define Residential College District's The Fresh Market as j; so j = q = 14. Define Ice Cream's Pineapple as X; so X = 2 + u = 2 + 0 = 2. Define Ice Cream's Banana as K; so K = C + X = 0 + 2 = 2. Define The Fresh Market's Ice Cream as P; i = j- f = 14 - 14 = 0; so P = 13 + i = 13 + 0 = 13. Define Jungle Jim's International Market's Ice Cream as R; so R = K- u = 2 - 0 = 2. Define School District's Jungle Jim's International Market as V; so V = P = 13. Define Jungle Jim's International Market's Cheese as v; so v = G + P = 0 + 13 = 13. Define Jungle Jim's International Market's Parmesan Cheese as S; so S = X = 2. Define Jungle Jim's International Market's Product as y; U = S + R = 2 + 2 = 4; so y = U + v = 4 + 13 = 17. Define School District's Product as J; so J = V * y = 13 * 17 = 14. Answer: 14

A number of features of these problems are immediately apparent: The problems are nonsense. The sentences all have the same structure. The arithmetic in the answers is carried out modulo 23.

2.1 Nonsense

As written, these problems are gibberish. The authors remark in a footnote, "We use simple English sentence templates to describe the problem, and did not worry about grammar mistakes such as singular vs plural forms," but the problems with a sentence like "The number of each The Fresh Market's Ice Cream equals 13 more than the difference of each Residential College District's The Fresh Market and each Parmesan Cheese's Grape" are not fixed by pluralizing "Ice Cream", "Market"' and "Grape". Since it is perfectly to write code that generates sentences of these kinds that are grammatically valid and individually sound perfectly reasonable, one can only suppose that the authors didn't at all care about it.

Another example in the paper, not from their data set and thus not constrained by their method of synthesis, shows similar problems: "Alice's apple is three times the sum of Bob's orange and Charles's banana." This is entirely meaningless as written. Even when writing examples in a completely unconstrained setting, the authors seem to be entirely uninterested in the realities of the English language.

2.2 Uniformity

All the sentences in the formulation questions follow a very limited template: "The number of ⟨term⟩ is equals ⟨arithmetic combination of terms⟩", where each atomic term has the form "each X's Y". Likewise, all the sentences in the solution follow a single template. Third, the "easy" example is extremely long, and the "hard" example is appallingly long.

The authors have not published their dataset, but they do provide the pseudocode for generating sentences (table 1), confirming the very limited class of sentences that the dataset contains.

The authors at one point boast: "**Proposition 2.2.** Ignoring unused parameters, numerics, sentence orderings, English words, a-z and A-Z letter choices, iGSM-med$^{op=15}$ [i.e. problems that involve 15 arithmetic operators] still has at least 7 billion solution templates, and iGSM-hard$^{op=21}$ [problems with 21 operations] has at least 90 trillion solution templates." But when one is dealing with combinations of 15 or 21 elements, 7 billion or 90 trillion are small numbers; they correspond to a branching factor of about $(7 \cdot 10^9)^{1/15} \approx 3.9$. As one point of contrast, the number of 15-word-long English sentences over a typical adult vocabulary that are both syntactically and semantically valid is certainly greater than 10^{50}. The number of such sentences that express numerical facts about collections of entities involving natural numbers no larger than 100 is certainly greater than 10^{48}.

Thus, the claim that their dataset is unusually diverse is the exact reverse of the truth; it is extraordinarily uniform. The authors seem to be unaware what actual linguistic diversity is like.

GenSentence(G_d, a)
1: str ← "The number of [name of a] equals"
2: pool ← { b ∈ G_d : ∃ b → a ∈ G_d }.
3: **if** RNG ∈ pool **then**
4: str ← str + "[random int between 0 and 22]"; and pool ← pool \ { RNG }
5: **If** |pool| > 0, str ← str + " more than" or " times" each with probability 0.5.
6: **if** |pool| = 1 then
7: str ← str + " [name of b]" for pool = b.
8: **else if** |pool| = |{b, c}| = 2 then
9: str ← str + " the sum of [b] and [c]" or " the difference of [b] and [c]" each w.p. 0.5.
10: **else**
11: str ← str + " the sum of .., .., and .." with a random order of all elements from pool.

Table 1: Pseudocode for generating sentence. The argument G_e is a dependency graph for the problem structure, and the argument a is an entity in that network.

2.3 Arithmetic mod 23

If the reader of this review looked carefully at the answer to the "easy" problem, they may have been puzzled by the claims that 17+13=7; that 12+20=9; and that 7*22=16. The explanation is simple; all the calculations been carried out modulo 23. The authors' justification for this odd choice is that it "avoids errors from computation involving large numbers". No doubt this does make the arithmetic easier to learn. There are, after all, only 529 different additions, 529 different subtractions, and 529 different multiplications when one is computing modulo 23, so no doubt the training set contains each of these many times. (How many times exactly I can't say, because I can't find any statement in this paper of how large the training set was. I can hardly believe that the authors omitted this critical statistic from this careful report on an experiment, so perhaps I have overlooked it, but I really can't find it.)

Given the way that they are generating problems, the near certainty of generated large numbers in the solution is, indeed, a serious concern. If you start with numbers between 0 and 22 and then throw addition, subtraction, and multiplication at them completely at random 21 times — which is, effectively, what they are doing — and you use ordinary arithmetic on integers rather than arithmetic mod 23, then the numbers are apt to get large. You will reasonably often generate numbers with 10 digits and occasionally with 20, and LLMs (and people) are error-prone on those kinds of arithmetic problems.

An alternative approach, of course, would be to generate problems that are more grounded in reality and therefore do not have answers like "The number of each The Fresh Market's Ice Cream equals 48,173,609,576" when carried out in ordinary arithmetic. This is easily done: Assign to every parameter in the problems a plausibly small numerical value (between 1 and 50, say), and then write the arithmetic constraints so that they correspond to the assigned number. However, the authors preferred instead to ground their word problems in a universe in which, if there are 7 children and each child has 20 cookies, there are a total of 2 cookies.[1]

The decision to use arithmetic modulo 23 not only makes it impossible for the calculations to have any real-world relevance; it also impacts the kinds of mathematics that can be done. In particular it is impossible to have any order relation that interacts in the standard way with arithmetical operations.

[1] A more cheerful formulation of the same proposition: If you have 2 cookies and you divide them equally among 7 children, then each child will get 20 cookies.

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?

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?

Problem: Tina buys 3 12-packs of soda for a party. Including Tina, 8 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 once the party is over?

Table 2: Problems from GSM8K (Cobbes et al. 2021)

3 Comparison to standard math word problems

As a contrast, let me point out the first three examples of GSM8K in (Cobbe et al. 2021) shown in Table 2. These are completely different. They are readable and fairly meaningful. They are much shorter. They involve a much richer vocabulary. Though not particularly exciting or ornate linguistically, they display a range of grammatical features.

Another deeper difference is that the problems from GSM8K draw significantly on commonsense knowledge. They are all narratives in which things change over time. In the second problem, the human reader must understand that when Mrs. Lim sells milk, the amount of milk she has decreases and the amount of money she has increases. The reader must also use their commonsense to determine what numbers to ignore; for instance, the fact that Mrs. Lim sells m gallons of milk for d dollars a gallon, then the number of gallons of milk she has decreases by m and her wealth in dollars increases by md. Common sense is also needed to determine which numbers to ignore; for instance the fact that Mrs. Lim milks her cows twice a day does not mean that the answer should be multiplied by 2.

In fact, Ye et al. state that avoiding problems that require commonsense was a deliberate choice of theirs to avoid the difficulties that commonsense reasoning entails. Whether this choice is reasonable depends on what the experiments are intended to accomplish. I will return to this point in section 6.

4 The claimed results

We can now review the results that Ye et al. claim to have determined from their experiment

Result 1 claims that the dataset is "highly diverse". I have discussed this in section 2.2 and demonstrated that it is the reverse of the truth. In fact, their dataset is much more uniform than any benchmark collection of word problems that is in use.

Result 2: "We demonstrate that the GPT2 model, pretrained on our synthetic dataset, not only achieves 99% accuracy in solving math problems from the same distribution but also out-of-distribution generalizes, such as to those of longer reasoning lengths than any seen during training." Considering the known power of the transformer architecture, the very limited number of possible arithmetic operations (a total of 1587 possible binary arithmetic problems), and the very narrow range of possible sentence forms, a 99% accuracy rate is not very surprising.

The claim about out-of-distribution generalization is based on the fact that their model was trained on

problems involving 15 arithmetic operations and it was able to successfully solve problems with 21 arithmetic operations. Technically, therefore, the test set was "out of distribution" relative to the training set. However, since the two sets were identical in all respects other than problem length, that is not a very impressive accomplishment in terms of generalization. If you train a classifier to distinguish images of cats from toasters using a training set with those two categories, and then determine that the classifier also succeeds in distinguishing images of dogs from blenders, then, similarly, the test set is "out of distribution" relative to the training set, but the result is not particularly impressive.

Result 3: "The model can learn to generate shortest solutions, almost always avoiding *unnecessary* computations." [Here, and in all the subsequent quotations from the paper, the various forms of emphasis — italics, boldface, and exclamation points — are the authors'.] I have no criticism of this result. It seems to me genuinely supported by the experiment, accurately stated, and rather interesting.

Result 4: "We examine the model's internal states through probing, introducing six probing tasks to elucidate how the model solves math problems. For instance, we discover the model (mentally!) preprocesses the full set of necessary parameters before it starts any generation." It is certainly possible that the new probing methods represent a contribution. I have not examined them, and I am not an expert in that area. The determination that the model preprocesses the parameters before starting generation seems to me a necessary consequence of Result 3 and a weaker result. The system could not reliably output the shortest solutions (result 3) unless it had computed the entire solution before beginning generation (result 4).

Result 5: "Surprisingly, the model also learns unnecessary, yet important skills after pretraining such as all-pair dependency. Before any question is asked, it already (mentally!) computes with good accuracy which parameters depend on which, even though *some are not needed for solving the math problem.* Note that computing all-pair dependency is a skill not needed to fit all the solutions in the training data. To the best of our knowledge, this is the first evidence that a language model can **learn useful skills beyond** those necessary to fit its pretraining data." The claim that these skill are beyond those necessary to fit the data is based on a misconstrual of the task being learned in training. The transformer model is not being trained to solve the math problem; it is being trained to predict the next token. It is not at all surprising that knowing the dependencies between parameters would be useful for next-token prediction. It is well known that transformer models learn semantic associations between words, which can be reflected in word embeddings. This new result seems to me like a weak version of that well-established resuot. So the new result is not very surprising and not at all new.

Result 6: "We explain *why* mistakes are made." They don't. Their final conclusions in their discussion of this claim (their section 5, p. 11) are as follows:

- "Many reasoning mistakes made by the language model are systematic, stemming from **errors in its mental process,** not merely random from the generation process."
- "Some of the model's mistakes can be discovered by probing its inner states **even before the model opens its mouth** (i.e., before it says the first solution step)."

The first is characterizes, vaguely, the kinds of errors that are made, and the second gives a little information about the stage of the process when the error is made. Neither is an explanation.

Results 7 and 8: Result 7 is stated as "Language model depth is crucial for mathematical reasoning." Result 8 elaborates this: "The depth of a language model is crucial, likely due to the complexity of its hidden (mental) reasoning processes. A t-step mental reasoning . . . may require deeper models for larger t, assuming all other hyperparameters remain constant."

If this were true, the consequences would be startling; it would seem to imply a transformer models of reasonable fixed depth cannot be used to generate any output that requires long reasoning chains (e.g. mathematical proofs, scientific analyses, legal briefs). However, they immediately state a disclaimer "[T]he above claim does not imply that "a t-step mental thinking requires a depth-t transformer. It is plausible for

a single transformer layer (containing many sub-layers) to implement t > 1 mental thinking steps, though possibly with reduced accuracy as t increases." It seems to me that those disclaimers entirely vitiate the point of result 7 as regards the way that actual LLMs are used in practice. I do not see why result 7, thus qualified, is of any interest or importance.

5 A transformer, not a large language model

I would further claim that the system they have studied is a trained transformer, but not a large language model. It is not any kind of language model at all. A large language model is a transformer that has been trained on a very large body of highly diverse text. A transformer that has been trained exclusively on some number of word problems that, between them, express nothing but elementary arithmetical constraints over values with no semantic content beyond those constrains, using a vocabulary of a few hundred words, mostly proper nouns, and a half-dozen or so different sentence structures, is not a language model.

The point is not merely one of terminology. The system studied in Ye et al. is not the same kind of thing as GPT-2, just with a different training set. In terms of its abilities and weaknesses, it is hardly any closer to GPT-2 than a 1990's style k-gram model. The training set is as important, or more, than the underlying architecture in determining the behavior of these systems.

It is not at all difficult to write a problem generator that generates a collection of word problems and answers that are written in idiomatic natural language; reasonably plausible as statements about the world, if not as epistemic states;[2] exhibit a significant range of linguistic form; and can be computed using standard arithmetic at no point requiring numbers larger than 100. If they are truly interested in large language models and mathematical word problems, they should have taken the trouble to do this.

6 Am I completely missing the point?

I will undoubtedly receive the response that I have completely missed the point of the paper I have reviewed here. This paper is Part 2.1 of a large-scale project whose overall purpose, as described on the project's website,[3] is to study the capacity of AI architectures in a controlled idealized environment, following in the hallowed footsteps of Newton, Kepler, and Tycho Brahe. My complaints above, that the problems are not realistic examples of math word problems, are of course true but completely irrelevant. The purpose of the experiments was not intended to study LLMs in a realistic setting; it was to study them in a controlled, perfectly understood, and abstract setting. The first paper in this series (Allen-Zhu and Li, 2023) does exactly that; it studies the ability of transformer models to learn extremely complex, abstract, context-free grammars.

However, if that is the purpose of the experiments and that is the proper perspective from which to analyze this paper, then why formulate it in terms of word problems? For most of us who study the workings of AI systems on math word problems, the way that those systems deal with the actual problems of language and with incorporating real-world knowledge is the most interesting aspect of the domain. If you are content to ignore the characteristics of natural language, and content to ignore the real-world grounding of arithmetic problems to the point that doing arithmetic modulo 23 seems reasonable, then why not carry out the experiments using a training experiment set with abstract arithmetical equations computed mod 23?

[2] Ideally, a math problem should be constructed so that it is plausible that a person might be cognizant of the facts in the problem statement; ignorant of the answer to the question; and interested in computing the answer to the question. The examples in table 2 satisfy this condition fairly well; a statement like, "The number of cookies that Mary has eaten this week is 3 times the number of Fred's first cousins plus 2" does not satisfy this epistemic constraint, though it may well be true as a statement of fact. However, problems in standard collections do not always satisfy this constraint, so this is not a very grave failing in the dataset in Ye et al.

[3] `https://physics.allen-zhu.com/`

Fundamentally, the data set that has been created amounts to nothing more. If the paper had been entitled, "An Analysis of the Capacity of a Transformer Model over Systems of Arithmetic Constraints Computed Modulo 23" and the contents of the paper were presented from that point of view, I would have not have written this long critique.

Acknowledgements

Thanks to Kevin Sullivan for pointing out the paper under review and to Devdatt Dubhashi for pointing out (Allen-Zhu and Li, 2023).

References

Allen-Zhu, Zeyuan and Yuanzi Li (2023). "Physics of Language Models: Part 1, Learning Hierarchical Language Structures." arXiv preprint arXiv:2305.13673 https://arxiv.org/pdf/2305.13673

Cobbe, Karl et al. (2021). "Training verifiers to solve math word problems." arXiv preprint arXiv:2110.14168 https://arxiv.org/abs/2110.14168

Hewitt, John and Christopher D. Manning (2019). A structural probe for finding syntax in word representations. NAACL-2019. https://aclanthology.org/N19-1419

Mishra, Swaroop, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit et al. (2022). "Lila: A unified benchmark for mathematical reasoning." arXiv preprint arXiv:2210.17517 https://arxiv.org/abs/2210.17517

Patel, Arkil, Satwik Bhattamishra, and Navin Goyal (2021)."Are NLP Models really able to Solve Simple Math Word Problems?." arXiv preprint arXiv:2103.07191 https://arxiv.org/abs/2103.07191

Ye, Tian, Zicheng Xu, Yuanzhi Li, and Zeynan Allen-Zhu (2024). "Physics of Language Models: Part 2.1, Grade-School Math and the Hidden Reasoning Process." arXiv preprint 2407.20311. https://arxiv.org/abs/2407.20311

Zhang, Hugh et al. (2024). A careful examination of large language model performance on grade school arithmetic. arXiv preprint arXiv:2405.00332. https://arxiv.org/abs/2405.00332