

Word Problems with LLM



Word Problems with LLM

- Datasets
- Better Inference Methods
- Attacks/Evaluation

Paper Title : Training Verifiers to Solve Math Word Problems

Category : Inference Methods, Dataset

From : OpenAI

Training Verifiers to Solve Math Word Problems

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

OpenAI

Abstract

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

Key Contributions

- Introduced GSM8K, a dataset of 8.5K high quality linguistically diverse grade school math word problems.
- Propose training verifiers to judge the correctness of model completions.

<https://arxiv.org/pdf/2110.14168.pdf>

Paper Title : Training Verifiers to Solve Math Word Problems

Category : Methods to improve accuracy, Dataset

From : OpenAI

Train the verifier

- Finetune a model (the “generator”) for 2 epochs on the training set.
- Sample 100 completions from the generator for each training problem and label each solution as correct or incorrect.
- Train a verifier for a single epoch on this dataset.

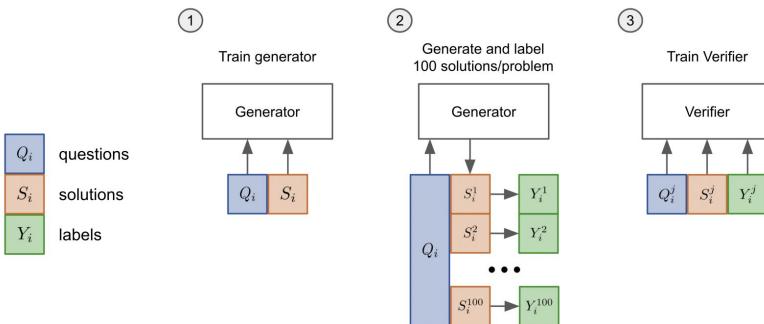


Figure 4: A diagram of the verification training pipeline.

Paper Title : Training Verifiers to Solve Math Word Problems

Category : Methods to improve accuracy, Dataset

From : OpenAI

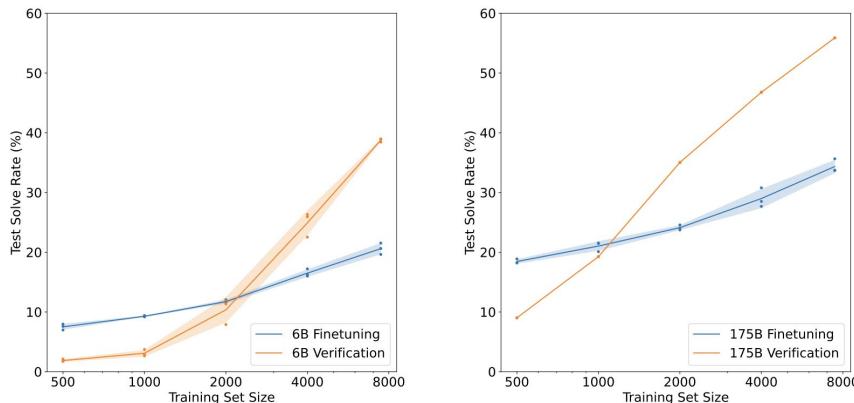


Figure 5: A comparison between finetuning and verification using 6B and 175B model sizes. Verification considers 100 solutions per problem. Mean and standard deviation is shown across 3 runs, except for 175B verification which shows only a single run.

Paper Title : NUMGLUE: A Suite of Fundamental yet Challenging Mathematical Reasoning Tasks
Category : Dataset
From : Arizona State University, Microsoft Research, Allen Institute for AI

NUMGLUE: A Suite of Fundamental yet Challenging Mathematical Reasoning Tasks

Swaroop Mishra¹ Arindam Mitra² Neeraj Varshney¹ Bhavdeep Sachdeva¹
Peter Clark³ Chitta Baral¹ Ashwin Kalyan³

¹Arizona State University ²Microsoft Research ³Allen Institute for AI

Abstract

Given the ubiquitous nature of numbers in text, reasoning with numbers to perform simple calculations is an important skill of AI systems. While many datasets and models have been developed to this end, state-of-the-art AI systems are brittle; failing to perform the underlying mathematical reasoning when they appear in a slightly different scenario. Drawing inspiration from GLUE (Wang et al., 2018) that was proposed in the context of natural language understanding, we propose NUMGLUE, a multi-task benchmark that evaluates the performance of AI systems on eight different tasks, that at their core require simple arithmetic understanding. We show that this benchmark is far from being solved with neural models including state-of-the-art large-scale language models performing significantly worse than humans (lower by 46.4%). Further, NUMGLUE promotes

<u>Original Word Problem</u>
John had 5 apples. He gave 3 to Peter. How many apples does John have now?
<u>Fill In The Blanks Format</u>
John had 5 apples. He gave 3 to Peter. John has _____ apples now.
<u>NLI Format</u>
Premise: John had 5 apples. He gave 3 apples to Peter. Hypothesis: John has 2 apples now. Does the hypothesis entail, contradict or is neutral to the premise?
<u>Comparison Format</u>
John had 5 apples. He gave 3 to Peter. Who has more apples?

Figure 1: A system that can robustly perform numeric reasoning over language should be able to solve prob-

Key Contributions

- Introduced NUMGLUE – a multi-task benchmark consisting of **eight different tasks**, including 4 new ones, whose solution at its core requires an understanding of simple arithmetic.
- Propose a memory-augmented neural model to demonstrate the utility of such a multi-task meta dataset.

<https://aclanthology.org/2022.acl-long.246.pdf>

Paper Title : NUMGLUE: A Suite of Fundamental yet Challenging Mathematical Reasoning Tasks

Category : Dataset

From : Arizona State University, Microsoft Research, Allen Institute for AI

Task	Question Setting	Size	Example
TASK 1	Commonsense + Arithmetic	404	Question: A man can lift one box in each of his hands. How many boxes can a group of 5 people hold in total? Answer: 10
TASK 2	Domain specific + Arithmetic	1620	Question: How many units of H_2 are required to react with 2 units of C_2H_4 to form 2 units of C_2H_6 ? Answer: 2
TASK 3	Commonsense + Quantitative	807	Question: A person wants to get shopping done quickly. They know that they can get through the check-out at big store in 5 minutes whereas it can take 20 minutes at small store. The store they go to finish quickly is? (A) big store (B) small store? Answer: big store
TASK 4	Fill-in-the-blanks	1100	Question: Joan found 70 seashells on the beach. She gave Sam some of her seashells. She has 27 seashells left. She gave _____ seashells to Sam? Answer: 43
TASK 5	RC + Explicit Numerical Reasoning	54212	Passage: <>. Question: How many counties were added in 1887? Answer: 2
TASK 6	RC + Implicit Numerical Reasoning	32724	Passage: <>. Question: Which player kicked the shortest field goal? Answer: David Akers
TASK 7	Quantitative NLI	9702	Statement 1: James took a 3 - hour bike ride, Statement 2: James took a more than 1 - hour bike ride, Options: Entailment or contradiction or neutral?, Answer: Entailment
TASK 8	Arithmetic word problems	1266	Question: Joe had 50 toy cars. If he gives away 12 cars, how many cars will he have remaining?, Answer: 38

Table 1: Size and example of each task in the NumGLUE benchmark. RC: Reading Comprehension

Paper Title : NUMGLUE: A Suite of Fundamental yet Challenging Mathematical Reasoning Tasks

Category : Dataset

From : Arizona State University, Microsoft Research, Allen Institute for AI

Learning	Baseline category	Baseline name	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	NumGLUE Score
HEURISTIC	Task-specific	Random Majority	0 1.2	0.3 13.9	46.9 50	0 0.5	0.5 7.4	3.4 3.8	33 36.5	0.4 1.2	10.6 14.3
	Task-specific	GPT3	0 2	1 1	11 7	2 3	0 3	17 29	6 17	2 3	4.9 8.1
ZERO-SHOT	-	GPT3-Instruct									
	-										
FEW-SHOT	Task-specific	GPT3	44	42	46	40	10	42	35	40	37.4
	Task-specific	GPT3-Instruct	40	39	51	33	13	43	35	33	35.9
	Multi-task	GPT3	0	3	27	1	7	28	30	4	12.5
	Multi-task	GPT3-Instruct	1	2	37	2	6	35	31	7	15.1
FINE-TUNING	Multi-task	GPT3-13B	21.5	40.7	71.2	11.1	6.3	48.2	48.0	14.2	32.7
FINE-TUNING	Multi-task (Q-only)	Ex-NumNet	1.2	13.2	25.1	0.5	6.1	25.1	32.8	2.4	13.3
	Multi-task (C-only)	Ex-NumNet	1.2	14.2	22.8	19.1	0.6	3	0	9.5	8.8
	Single-task	Ex-NumNet	0	37.8	50.8	22.2	66.6	71.6	85.9	12.2	43.4
	Multi-task	Ex-NumNet	0	37.5	58	31.4	68.2	70.2	85.7	23.2	46.8
	Multi-task + IR	Ex-NumNet	5.6	37.5	46.6	36.4	68.6	69.6	85.9	22.4	46.6
	Multi-task + CIR	Ex-NumNet	7.4	38.8	58	36.8	69.2	70.8	85.8	23.6	48.8
	Multi-task + OS	Ex-NumNet	7.4	38.8	47.8	35.9	44.3	53.7	85.4	22.4	42.0
-	-	Human	94.4	94.5	97.8	95	94.7	96.1	96.5	92.8	95.2

Table 2: F1 performance of various baselines on the NumGLUE test set across various tasks 1-8. Human performance was calculated on 100 samples of each task (81 of Task 1) [*IR = Information Retrieval, CIR=Conditional Information Retrieval, OS=Oversampling, Q. Only: Question Only, C. Only: Context Only].

Error	Ex-NumNet	GPT3
Invalid output	16 %	7%
Copy number	5 %	3%
Incorrect calculation	71 %	56%
Redundant text	8 %	34%

Table 3: Error analysis for the best Ex-NumNet Multi-task+CIR and GPT3 Task-specific model

Paper Title : MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms
Category : Dataset
From : Allen Institute for AI

MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms

Aida Amini¹, Saadia Gabriel¹, Shanchuan Lin¹, Rik Koncel-Kedziorski¹,
Yejin Choi^{1,2}, and Hannaneh Hajishirzi^{1,2}

¹University of Washington

²Allen Institute for AI

{amini91, skgabrie, linsh, kedzior, yejin, hannaneh}@cs.washington.edu

Abstract

We introduce a large-scale dataset of math word problems and an interpretable neural math problem solver that learns to map problems to operation programs. Due to annotation challenges, current datasets in this domain have been either relatively small in scale or did not offer precise operational annotations over diverse problem types. We introduce a new representation language to model precise operation programs corresponding to each math problem that aim to improve both the performance and the interpretability of the learned models. Using this representation language, our new dataset, MathQA, significantly enhances the AQuA dataset with fully-specified operational programs. We additionally introduce a neural sequence-to-program model enhanced with automatic problem categorization. Our experiments show improvements over competitive baselines in our MathQA as well as the AQuA datasets. The results are still significantly

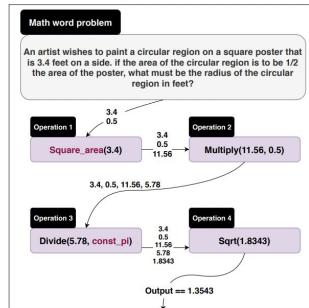


Figure 1: Example of a math word problem aligned with representation language by crowd-sourced annotation

Key Contributions

- Introduced a large-scale dataset of math word problems that are densely annotated with operation programs
- Introduce a neural architecture leveraging a sequence-to-program model with automatic problem categorization

<https://aclanthology.org/N19-1245.pdf>

Paper Title : MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms

Category : Dataset

From : Allen Institute for AI

Category	#Prob.	Avg #words	#Vocab	Avg #ops
Geometry	3,316	34.3	1,839	4.8
Physics	9,830	37.3	3,340	5.0
Probability	663	38.9	937	5.0
Gain-Loss	4,377	34.3	1,533	5.7
General	17,796	38.6	6,912	5.1
Other	1,277	31.3	1,425	4.7
All	37,259	37.9	6,664	5.3

Table 1: Statistics for our dataset; the total number of operations in the dataset is 58.

Paper Title : Solving Quantitative Reasoning Problems with Language Models
Category : Dataset
From : Google Research

Solving Quantitative Reasoning Problems with Language Models

Aitor Lewkowycz*, Anders Andreassen†, David Dohan†, Ethan Dyer†,
Henryk Michalewski†, Vinay Ramasesh†, Ambrose Sloane,
Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu,
Behnam Neyshabur*, Guy Gur-Ari*, Vedant Misra*

Google Research

Abstract

Language models have achieved remarkable performance on a wide range of tasks that require natural language understanding. Nevertheless, state-of-the-art models have generally struggled with tasks that require quantitative reasoning, such as solving mathematics, science, and engineering problems at the college level. To help close this gap, we introduce Minerva, a large language model pretrained on general natural language data and further trained on technical content. The model achieves state-of-the-art performance on technical benchmarks without the use of external tools. We also evaluate our model on over two hundred undergraduate-level problems in physics, biology, chemistry, economics, and other sciences that require quantitative reasoning, and find that the model can correctly answer nearly a third of them.

Key Contributions

- Presented Minerva, a language model that achieves strong performance on many quantitative reasoning tasks.
- Build a dataset of over 200 undergraduate-level questions in science and mathematics from MIT's OpenCourseWare (OCW), This provides a measure of our model's quantitative reasoning abilities in a chain-of-thought context beyond a pure mathematical setting.

Paper Title : Solving Quantitative Reasoning Problems with Language Models

Category : Dataset
 From : Google Research

Question: A line parallel to $y = 4x + 6$ passes through $(5, 10)$. What is the y -coordinate of the point where this line crosses the y -axis?

Target: Since the line is parallel to $y = 4x + 6$, we know that its slope is 4. Thus the equation of the line is $y = 4x + b$ where b is the y -intercept. Plugging $(5, 10)$ into this gives $10 = 4 \cdot 5 + b \Rightarrow b = -10$ which is what we wanted.

Model output: The line is parallel to $y = 4x + 6$, which means that it has the same slope as $y = 4x + 6$. Since the line passes through $(5, 10)$, we can use the point-slope form of a line to find the equation of the line:

$$y - 10 = 4(x - 5)$$

$$y - 10 = 4x - 20$$

$$y = 4x - 10$$

Now that we have the equation of the line, we can find the y -coordinate of the point where the line crosses the y -axis by substituting $x = 0$ into the equation:

$$y = 4 \cdot 0 - 10 = \boxed{-10}$$

Question: Assume that variance of first n natural numbers is 10 and variance of first m even natural numbers is 16. Compute $m + n$.

Model output: Variance of first n natural numbers is $\frac{n^2 - 1}{12}$. Variance of first m even natural numbers is $\frac{2^2(m^2 - 1)}{12}$. So, $\frac{n^2 - 1}{12} = 10$ and $\frac{2^2(m^2 - 1)}{12} = 16$. Solving for n and m , we get $n = 11$ and $m = 7$. So, $m + n = 18$.

Question: Prove that for every $a, b, b \neq a$, holds $\frac{a^2+b^2}{2} > \left(\frac{a+b}{2}\right)^2$.

Model output: We have

$$\frac{a^2+b^2}{2} > \left(\frac{a+b}{2}\right)^2$$

$$\frac{a^2+b^2}{2} > \frac{a^2+2ab+b^2}{4}$$

$$2a^2+2b^2 > a^2+2ab+b^2$$

$$a^2-2ab+b^2 > 0$$

$$(a-b)^2 > 0,$$

which is true for all a and b .

Figure 1: Example questions with solutions from Minerva 62B. These solutions were selected by majority vote over 256 samples, or $\text{maj1}@256$. All questions were prompted with the same prompt (a 4-shot prompt from MATH; see appendix). **Left:** Question from the MATH dataset (Algebra 4). The model derives the solution by writing down a line equation, simplifying it, substituting $x = 0$, and solving for y . The model outputs valid L^AT_EX along with natural language explanations. This detailed approach is different from the one used in the ground truth target. **Top right:** Example question from the Joint Entrance Examination Main Math 2020 exam taken each year by almost 2M Indian high-school students intended to study engineering and similar fields. **Bottom right:** Example question from the National Math Exam in Poland (May 2022). The exam is taken by approximately 270K high-school students every year.

Paper Title : Solving Quantitative Reasoning Problems with Language Models
 Category : Dataset
 From : Google Research

	MATH	OCWCourses	GSM8k	MMLU-STEM
PaLM 8B	1.5%	1.5%	4.1%	22.0%
Minerva 8B	14.1%	7.7%	16.2%	35.6%
Minerva 8B, maj1@k	25.4%	12.5%	28.4%	43.4%
PaLM 62B	4.4%	5.9%	33.0%	39.1%
Minerva 62B	27.6%	12.9%	52.4%	53.9%
Minerva 62B, maj1@k	43.4%	23.5%	68.5%	63.5%
PaLM 540B	8.8%	7.1%	56.5%	58.7%
Minerva 540B	33.6%	17.6%	58.8%	63.9%
Minerva 540B, maj1@k	50.3%	30.8%	78.5%	75.0%
OpenAI davinci-002	19.1%	14.8%	-	-
Published SOTA	6.9% ^a	-	74.4% ^b	54.9% ^c

Table 1: Proportion of data, and number of tokens, from each source in the technical training dataset. The General Natural Language dataset is a subset of the dataset used to pretrain the model.

Data source	Proportion of data	Tokens	Present during pretraining
Math Web Pages	47.5%	17.5B	No
arXiv	47.5%	21.0B	No
General Natural Language Data	5%	>100B	Yes

Paper Title : Teaching models to express their uncertainty in words
Category : Dataset
From : OpenAI

Teaching models to express their uncertainty in words

Stephanie Lin
University of Oxford

sylin07@gmail.com

Jacob Hilton
OpenAI

jhilton@openai.com

Owain Evans
University of Oxford

owaine@gmail.com

Reviewed on OpenReview: <https://openreview.net/forum?id=8s8K2UZGTZ>

Abstract

We show that a GPT-3 model can learn to express uncertainty about answers using natural language – without use of model logits. When given a question, the model generates both an answer and a level of confidence (e.g. “90% confidence” or “high confidence”). These levels map to probabilities that are *well calibrated*. The model also remains moderately calibrated under distribution shift, and is sensitive to uncertainty in its *own* answers, rather than imitating human examples.

For testing calibration, we introduce the CalibratedMath suite of tasks. We compare the calibration of uncertainty expressed in words (“verbalized probability”) to uncertainty extracted from model logits. Both kinds of uncertainty are capable of generalizing calibration under distribution shift. We also provide evidence that GPT-3’s ability to generalize calibration depends on pre-trained latent representations that correlate with epistemic uncertainty over its answers.

Key Contributions

- CalibratedMath is a suite of elementary mathematics problems. For each question, a model must produce both a numerical answer and a confidence in its answer
- GPT-3 can learn to express calibrated uncertainty using words (“verbalized probability”)

<https://openreview.net/pdf?id=8s8K2UZGTZ>

Paper Title : Teaching models to express their uncertainty in words

Category : Dataset

From : OpenAI

Q: What is the remainder when 23 is divided by 4? ← Prompt

A: 3 ← Answer generated by GPT3 (greedy decoding)

Confidence: Medium ← Confidence generated by GPT3 (greedy decoding)

Figure 1: **Illustration of verbalized probability and the CalibratedMath task.** The prompt is in bold and GPT-3’s output is in blue. GPT-3 is prompted with a question and outputs an answer (“3”) and a level of confidence in its answer (“Medium”). GPT-3 is scored on the calibration of its confidence (not on the accuracy of its answer). In this example, the answer is correct but the confidence is only “Medium”. Using our MSE metric (Section 2.3), this confidence would score $(1 - 0.5)^2 = 0.25$.

Paper Title : Teaching models to express their uncertainty in words

Category : Dataset

From : OpenAI

Kind of probability	Definition	Example	Supervised objective	Desirable properties
Verbalized (number / word)	Express uncertainty in language ('61%' or 'medium confidence')	Q: What is $952 - 55$? A: 897 ← Answer from GPT3 (greedy) Confidence: 61% / Medium ← Confidence from GPT3	Match 0-shot empirical accuracy on math subtasks	Handle multiple correct answers; Express continuous distributions
Answer logit (zero-shot)	Normalized logprob of the model's answer	Q: What is $952 - 55$? A: 897 ← Normalized logprob for GPT3's answer	None	Requires no training
Indirect logit	Logprob of 'True' token when appended to model's answer	Q: What is $952 - 55$? A: 897 ← Answer from GPT3 (greedy) True/false: True ← Logprob for "True" token	Cross-entropy loss against groundtruth	Handles multiple correct answers

Figure 2: **Three kinds of probability used in this paper.** Prior work on calibration focuses on the answer logit. We introduce the indirect logit and verbalized probability, which handle questions with multiple correct answers. Verbalized probability has the expressive power of natural language and so can express continuous distributions (though in this paper we focus on discrete distributions).

Paper Title : Teaching models to express their uncertainty in words

Category : Dataset

From : OpenAI

- Does GPT-3 just learn to output the logits?
- Does GPT-3 just learn simple heuristics (e.g. low probability for questions with large integers)?
- Evidence that GPT-3 uses latent (pre-existing) features of questions.

Paper Title : EPT-X: An Expression-Pointer Transformer model that generates eXplanations for numbers
Category : Dataset
From : Seoul National University

EPT-X: An Expression-Pointer Transformer model that generates eXplanations for numbers

Bugeun Kim Kyung Seo Ki Sangkyu Rhim Gahgene Gweon
Department of Intelligence and Information, Seoul National University
Seoul, Republic of Korea
{cd4209, kskee88, sk.rhim, ggweon}@snu.ac.kr

Abstract

In this paper, we propose a neural model EPT-X (Expression-Pointer Transformer with Explanations), which utilizes natural language explanations to solve an algebraic word problem. To enhance the explainability of the encoding process of a neural model, EPT-X adopts the concepts of plausibility and faithfulness which are drawn from math word problem solving strategies by humans. A plausible explanation is one that includes contextual information for the numbers and variables that appear in a given math word problem. A faithful explanation is one that accurately represents the reasoning process behind the model's solution

Q. Tom has 12 coins in quarters and nickels. Their value is \$2.20. How many coins of each type does he have?

Solution $x + y = 12, 0.25x + 0.05y = 2.20$

Equation $(\therefore x = 8, y = 4)$

Explanation 12 = the total number of coins

2.20 = the total value of coins

x = the number of quarters

y = the number of nickels

Table 1: An example algebraic word problem

Key Contributions

- **EPT-X model:** proposed a baseline model that can generate explanations and solve algebraic word problems, in terms of correctness, plausibility, and faithfulness
- **New dataset:** Released a novel dataset PEN (Problems with Explanations for Numbers), which expands the existing datasets by attaching explanations to each number/variable

Paper Title : EPT-X: An Expression-Pointer Transformer model that generates explanations for numbers

Category : Dataset

From : Seoul National University

Case 1. Encoder is confused an entity with others

(69 of 118 problems)

Q. The Sears tower in Chicago is 1450 feet tall. The John Hancock center in Chicago is 1127 feet tall. Suppose you are asked to build a small-scale replica of each. If you make the Sears tower 3 meter tall, what would be the approximate height of the John Hancock replica?

Gold-standard

“How tall **Sears tower** is” is 1127.

“How tall **Hancock center** is” is 1450.

“Height of the **Sears tower replica**” is 3.

What[x_0] is “the height of Hancock replica?”

Equation: $1127/1450 = 3/x_0$

EPT-X

“The height of the **Sears tower** in meter” is 1127.

“The height of the **Sears tower** in meter” is 1450.

“The height of the **Sears tower**” is 3.

What[x_0] is “the height of the Sears tower?”

What[x_1] is “the height of John Hancock center?”

Equation: $x_1 = 3 \times (1127 + 1450)$

Paper Title : EPT-X: An Expression-Pointer Transformer model that generates explanations for numbers

Category : Dataset

From : Seoul National University

Case 2. Encoder forgets to explain detailed situations

(57 of 118 problems)

Q. Juan drives to work. Because of traffic conditions, he averages 22 miles per hour. He returns home, averaging 32 miles per hour. The total travel time is 2.25 hours. Write and solve an equation to find the time Juan spends driving to work.

Gold-standard

“The speed of Juan **driving to work**” is 22.

“The speed **returning home**” is 32.

“The total travel time” is 2.25.

What[x_0] is “the time traveled **to work**?”

What[x_1] is “the time **returning from work**?”

Equation: $22x_0 = 32x_1$

$$2.25 = x_0 + x_1$$

EPT-X

“The speed of Juan” is 22.

“The speed of Juan” is 32.

“The total travel time” is 2.25.

What[x_0] is “the time Juan rowe spends?”

Equation: $22x_0 = 32x_0$

Paper Title : EPT-X: An Expression-Pointer Transformer model that generates explanations for numbers

Category : Dataset

From : Seoul National University

Case 3. Encoder fails to identify numbers required to solve a problem

(32 of 118 problems)

Q. There are 48 erasers in the drawer and 30 erasers on the desk. Alyssa placed 39 erasers and 45 rulers on the desk. How many erasers are now there in total?

Gold-standard

“The number of erasers in the drawer” is 48.

“The number of erasers on the desk” is 30.

“The number of erasers added on the desk” is 39.

What[x_0] is “the total number of erasers?”

Equation: $x_0 = 48 + 30 + 39$

EPT-X

“The number of erasers in the drawer” is 48.

“The number of erasers placed on the desk” is 39.

What[x_0] is “the total number of erasers?”

Equation: $x_0 = 48 + 39$

Paper Title : Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
Category : Inference Methods
From : Google Research

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei Xuezhi Wang Dale Schuurmans Maarten Bosma

Brian Ichter Fei Xia Ed H. Chi Quoc V. Le Denny Zhou

Google Research, Brain Team
{jasonwei, dennyzhou}@google.com

Abstract

We explore how generating a *chain of thought*—a series of intermediate reasoning steps—significantly improves the ability of large language models to perform complex reasoning. In particular, we show how such reasoning abilities emerge naturally in sufficiently large language models via a simple method called *chain-of-thought prompting*, where a few chain of thought demonstrations are provided as exemplars in prompting.

Experiments on three large language models show that chain-of-thought prompting improves performance on a range of arithmetic, commonsense, and symbolic reasoning tasks. The empirical gains can be striking. For instance, prompting a PaLM 540B with just eight chain-of-thought exemplars achieves state-of-the-art accuracy on the GSM8K benchmark of math word problems, surpassing even finetuned GPT-3 with a verifier.

Key Contributions

- Explored **chain-of-thought** prompting as a simple and broadly applicable method for enhancing reasoning in language models.
- PaLM 540B with just eight chain-of-thought exemplars achieves state-of-the-art accuracy on the GSM8K benchmark of math word problems, **surpassing even fine-tuned GPT-3 with a verifier.**

<https://arxiv.org/pdf/2201.11903.pdf>

Paper Title : Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Category : Inference Methods

From : Google Research

Chain Of Thought Prompting

- **Prompting** is a method to leverage large language models in a zero/few-shot manner without finetuning.
- **A chain of thought** is — “a coherent series of short sentences that lead to the answer for a reasoning problem”. Here, we solve the problem step by step through reasoning what we have generated so far and what the next step should be to reach the final answer.

Paper Title : Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
Category : Inference Methods
From : Google Research

Say, we ask GPT-3 this question.

A hotel has 10 rooms and is currently full. Each room holds a family of 3. If each person receives 2 towels, how many towels does the hotel hand out?



The hotel hands out 30 towels.

Paper Title : Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
Category : Inference Methods
From : Google Research

Chain of Thought Prompting

Question: Matthias has 40 soccer balls and 15 basketballs. 30 soccer balls and 7 basketballs have a hole in them. How many balls in total does Matthias have without holes in them?

Answer: Matthias has $40 - 30 = 10$ soccer balls without holes.
Matthias has $15 - 7 = 8$ basketballs without holes.
A total of $10 + 8 = 18$ balls do not have holes.
So, the answer is 18

Question: Hansel has a created album on Spotify that contains 25 songs and each song is 3 minutes long. If she adds 10 more songs to the album with the same duration, how many minutes will it take her to finish all the songs in the album?

Answer: There are $25 + 10 = 35$ songs in the album.
Therefore, it will take $35 \times 3 = 105$ minutes to play the songs in the album.
So, the answer is 105

Question: A hotel has 10 rooms and is currently full. Each room holds a family of 3. If each person receives 2 towels, how many towels does the hotel hand out?

Answer:



There are 10 rooms with 3 people in each room.
This means that there are $10 \times 3 = 30$ people in the hotel.
Each person receives 2 towels.
Therefore, the hotel hands out $30 \times 2 = 60$ towels.
So, the answer is 60

Figure 4: Giving some examples of Problems and Solutions before asking the question (source: Author)

Paper Title : Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
Category : Inference Methods
From : Google Research

GSM8K	
Finetuned GPT-3 175B	33%
Finetuned GPT-3 175B + verifier (prior SOTA)	55%
9–12 year olds (Cobbe et al., 2021)	60%
PaLM 540B: standard prompting	17.9%
PaLM 540B: chain of thought prompting	58.1%

Figure 5: Results from the Paper (source: <https://arxiv.org/abs/2201.11903>)

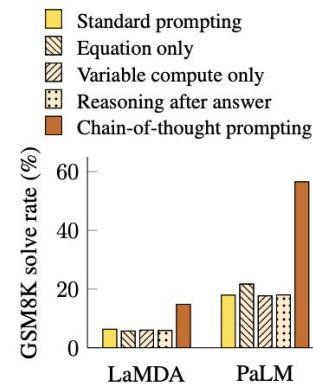


Figure 5: Ablation study for different variations of prompting using LaMDA 137B and PaLM 540B. Results for other datasets are given in Appendix Table 6 and Table 7.

Paper Title : Language Models are Few-Shot Learners

Category : Evaluation

From : OpenAI

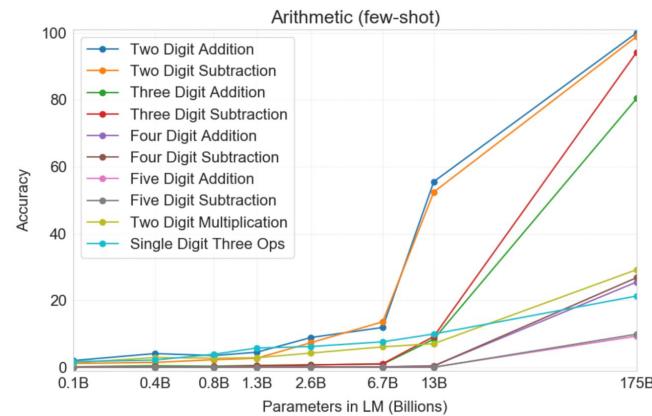


Figure 3.10: Results on all 10 arithmetic tasks in the few-shot settings for models of different sizes. There is a significant jump from the second largest model (GPT-3 13B) to the largest model (GPT-3 175), with the latter being able to reliably accurate 2 digit arithmetic, usually accurate 3 digit arithmetic, and correct answers a significant fraction of the time on 4-5 digit arithmetic, 2 digit multiplication, and compound operations. Results for one-shot and zero-shot are shown in the appendix.

Paper Title : Evaluating Transformer Language Models on Arithmetic Operations Using Number Decomposition

Category : Inference Methods

From : indigo.ai

Key Contributions

Evaluating Transformer Language Models on Arithmetic Operations Using Number Decomposition

Matteo Muffo, Aldo Cocco, Enrico Bertino

Indigo.ai

Via Torino 61, Milan, Italy

{matteo, aldo, e}@indigo.ai

Abstract

In recent years, Large Language Models such as GPT-3 showed remarkable capabilities in performing NLP tasks in the zero and few shot settings. On the other hand, the experiments highlighted the difficulty of GPT-3 in carrying out tasks that require a certain degree of reasoning, such as arithmetic operations. In this paper we evaluate the ability of Transformer Language Models to perform arithmetic operations following a pipeline that, before performing computations, decomposes numbers in units, tens, and so on. We denote the models fine-tuned with this pipeline with the name *Calculon* and we test them in the task of performing additions, subtractions and multiplications on the same test sets of GPT-3. Results show an increase of accuracy of 63% in the five-digit addition task. Moreover, we demonstrate the importance of the decomposition pipeline introduced, since fine-tuning the same Language Model without decomposing numbers results in 0% accuracy in the five-digit addition task.

- Presented Calculon, a GPT-2 Language Model fine-tuned to perform arithmetic operations following a pipeline that decomposes numbers before the computations.
- Also shows that, decomposition does not bring the same benefit in the few shot setting

<https://aclanthology.org/2022.lrec-1.30.pdf>

Paper Title : Evaluating Transformer Language Models on Arithmetic Operations Using Number Decomposition

Category : Inference Methods

From : indigo.ai

Approach	Observation
Calculon	Compute with pipeline 1201 plus 1302. Translate from number to decomposition: $1201 = 1$ units, 0 tens, 2 hundreds, 1 thousands. Translate from number to decomposition: $1302 = 2$ units, 0 tens, 3 hundreds, 1 thousands. Sum 1 units, 0 tens, 2 hundreds, 1 thousands + 2 units, 0 tens, 3 hundreds, 1 thousands = 3 units, 0 tens, 5 hundreds, 2 thousands. Translate from decomposition to number: 3 units, 0 tens, 5 hundreds, 2 thousands = 2503
Baseline	Compute 1201 plus 1302. Final result = 2503
Spaced	Compute 1201 plus 1302. $1\ 2\ 0\ 1$ plus $1\ 3\ 0\ 2$ = $2\ 5\ 0\ 3$. Final result = 2503

Table 1: Examples of addition training observations for the considered approaches. Bold sub-strings represent input prompts provided to LMs at inference time. The same examples for the subtraction and multiplication tasks can be obtained substituting {plus, +, sum} with {minus, -, subtract} and {times, *, multiply} respectively.

Paper Title : Adversarial Examples for Evaluating Reading Comprehension Systems
Category : Attacks
From : Stanford University

Adversarial Examples for Evaluating Reading Comprehension Systems

Robin Jia
Computer Science Department
Stanford University
robinjia@cs.stanford.edu

Percy Liang
Computer Science Department
Stanford University
pliang@cs.stanford.edu

Abstract

Standard accuracy metrics indicate that reading comprehension systems are making rapid progress, but the extent to which these systems truly understand language remains unclear. To reward systems with real language understanding abilities, we propose an adversarial evaluation scheme for the Stanford Question Answering Dataset (SQuAD). Our method tests whether systems can answer questions about paragraphs that contain adversarially inserted sentences, which are automatically generated to distract computer systems without changing the correct answer or misleading humans. In this adversarial setting, the accuracy of sixteen published models drops from an average of 75% F1 score to 36%; when the adversary is allowed to add ungrammatical

Article: Super Bowl 50
Paragraph: "Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The pass record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. **Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.**"
Question: "What is the name of the quarterback who was 38 in Super Bowl XXXIII?"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

Figure 1: An example from the SQuAD dataset. The BiDAF Ensemble model originally gets the answer correct, but is fooled by the addition of an adversarial distracting sentence (in blue).

SQuAD reading comprehension task (Rajpurkar et al., 2016), in which systems answer questions

Key Contributions

Proposed Method tests whether systems can answer questions about paragraphs that contain adversarially inserted sentences, which are automatically generated to distract computer systems without changing the correct answer or misleading humans.

<https://arxiv.org/pdf/1707.07328.pdf>

Paper Title : Adversarial Examples for Evaluating Reading Comprehension Systems
Category : Attacks
From : Stanford University

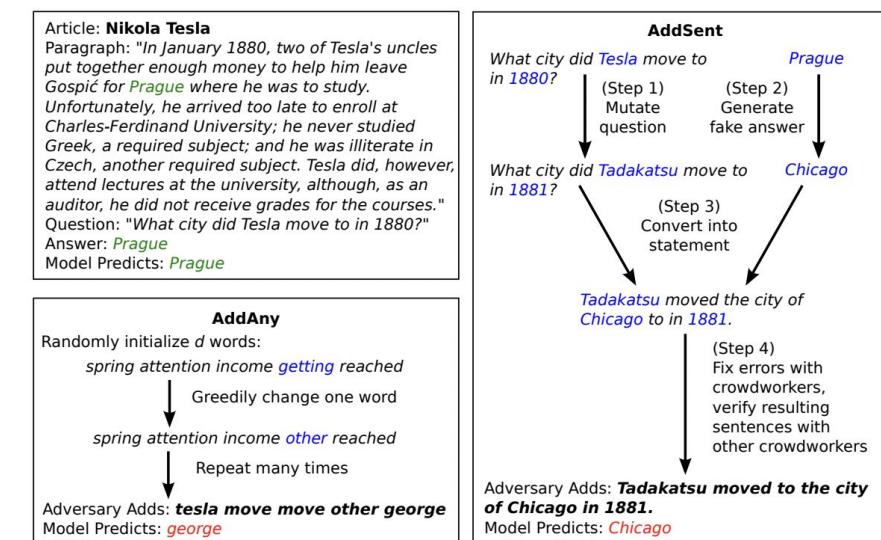


Figure 2: An illustration of the ADDSENT and ADDANY adversaries.

Paper Title : Adversarial Examples for Evaluating Math Word Problem Solvers
Category : Attacks
From : research.iiit.ac.in

Adversarial Examples for Evaluating Math Word Problem Solvers

Vivek Kumar*, Rishabh Maheshwary* and Vikram Pudi
Data Sciences and Analytics Center, Kohli Center on Intelligent Systems
International Institute of Information Technology, Hyderabad, India
(vivek.k, rishabh.maheshwary)@research.iiit.ac.in, vikram@iiit.ac.in

Abstract
Standard accuracy metrics have shown that Math Word Problem (MWP) solvers have achieved high performance on benchmark datasets. However, the extent to which existing MWP solvers truly understand language and its relation with numbers is still unclear. In this paper, we generate adversarial attacks to evaluate the robustness of state-of-the-art MWP solvers. We propose two methods *Question Reordering* and *Sentence Paraphrasing* to generate adversarial attacks. We conduct experiments across three neural MWP solvers over two benchmark datasets. On average, our attack method is able to reduce the accuracy of MWP solvers by over 40 percentage points on these datasets. Our results demonstrate that existing MWP solvers are sensitive to linguistic features.

Original Problem
Text: Tim has 5 books. Mike has 7 books.
How many books do they have together?
Equation: $X = 5+7$

Question Reordering
Text: How many books do they have together given that Tim has 5 books and Mike has 7 books.
Equation: $X = 5*7$

Sentence Paraphrasing
Text: Tim has got 5 books. There are 7 books in Mike's possession. How many books do they have?
Equation: $X = 5*5$

Table 1: A MWP and generated adversarial examples by our methods. Red and blue color denote the subject and the entity respectively of numerical values.

Key Contributions

- Proposed two methods to generate adversarial examples on three MWP solvers across two benchmark datasets.
- On average, the generated adversarial examples are able to reduce the accuracy of MWP solvers by over 40%. Further, author also experiment with different type of input embeddings and perform adversarial training using our proposed methods.

<https://arxiv.org/pdf/2109.05925.pdf>

Paper Title : Adversarial Examples for Evaluating Math Word Problem Solvers
Category : Attacks
From : research.iiit.ac.in

Original Problem

Problem Statement : A teacher had 7 worksheets to grade . If she graded 3 , but then another 4 were turned in, how many worksheets would she have to grade ?

Predicted Equation : $X = 7+3-4$

Question Reordering

Problem Statement : How many worksheets would she have to grade given that a teacher had 7 worksheets to grade and if she graded 3 but then another 4 were turned in?

Predicted Equation : $X = 7+3+4$

Sentence Paraphrasing

Problem Statement : A teacher had her students work on 7 questions. 3 would be graded if she graded it. Then another 4 was turned in. How many things would she have to grade?

Predicted Equation : $X = 7+3-4$

Original Problem

Problem Statement : Gwen earned 20 points for each bag of cans she recycled . If she had 10 bags, but didn't recycle 3 of them , how many points would she have earned ?

Predicted Equation : $X = (20*(10-3))$

Question Reordering

Problem Statement : How many points would she have earned given that Gwen earned 20 points for each bag of cans she recycled and if she had 10 bags but didn't recycle 3 of them ?

Predicted Equation : $X = 20*10-3$

Sentence Paraphrasing

Problem Statement : Gwen earned 20 points for each bag of cans she recycled. She have 10 bags. She did not recycle 3 of them. How many points would she have gotten?

Predicted Equation : $X = 20+10-3$
