HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 1 (2023)

Analyzing the GPT-3 AI's Ability to Predict the Answer to Algebraical Questions

Daniil Novak

Castro Valley High School

ABSTRACT

AI Algorithms have been getting increasingly smart and complex since their conception. The Most Modern AI's currently available are able to solve incredibly complex problems and are even able to solve abstract problems. It is only a matter of time until AI is able to supersede humans in Mathematical Calculations. (MGI et al. 2022). In this paper we took the most advanced AI currently available for commercial usage, GPT-3, and programmed it to use randomly generated mathematical input in order to test its solving abilities. By feeding this input, we evaluated whether the AIs ability to solve problems is solely based on complexity. We, however, found out that AI was also influenced by the type of operation. The distance and similarity metrics of many different AI Generated answers were taken and then compared with the actual ones. We conducted a series of T-tests to determine which results are statistically significant. We found out that the type of arithmetical expression in the problem, which was either addition, multiplication, division or subtraction, was significant (p=.0000027), suggesting that the type of problem matters more than previously expected.

INTRODUCTION

Computers and AI have been getting more and more advanced each year.Mozina and colleagues [16] demonstrated how computer models can use argumentation to learn how to solve problems. Computers are beginning to finally understand some human concepts, and with that, they are learning to solve human problems. AI takes in large amounts of data and finds the most correct result (Gkionak [6]).

All computers are driven by simple algebraic concepts, which allow them to function. Even our personal computers and cell phones have processors which have millions of microprocessors inside which use algebra to operate. It has allowed for the concept of machine intelligence to even exist in the first place.

The Basic structure of neural-network based AI consists of a set of interconnected mathematical expressions called neurons, which all work together in a large filtering process. Each expression has a number of input parameters and an output value which then can be used as an input for other neurons. To control and adjust the neuron expression each input is associated with a weight value. The outputs of the top-layer neurons represent the most probable or most correct response. These neurons use simple algebra to function and operate. To summarize: neurons in every AI usually follow an algebraic equation, where the weight and probability of an answer is determined.

The main tool for adjusting the quality of AI neural networks is called training. Training, roughly, is based on providing the set of problems with answers and then gradual adjusting of neuron parameters, weight values associated with input of each neuron so that the accuracy and precision of answers from the training set increases. This procedure is called backpropagation and is based on the gradual descent method for finding function optima. So, by tweaking the weight values we try to achieve the maximum performance evaluated as a formula based on the number of right and wrong answers from the problems from the training dataset.

As AI has become increasingly more complex, it has been able to grasp and solve modern concepts quicker than even Humans are able to. In the Article *What's Next For AI: Solving Advanced Math Problems* (Sagar, Ram[2]) the researchers wrote "Last year, Google researchers used machine learning algorithms to solve partial differential

equations that otherwise would take eons to solve. There was also this neural ODEs paper from late 2018, which showed promising results. The role of neural networks in the field of mathematics is slowly on the rise and can help researchers in finding solutions faster than usual. These models are made tireless, smart and can be leveraged to expose blind spots in the existing mathematical approaches.".

As the researchers said, AI is becoming increasingly complex with more parameters and more neurons which are able to solve randomly generated math equations more accurately and quickly. Computers and Machines are incredibly good at solving complex problems with incredibly specific equations and procedures. But what AI needs to be able to do is specifically detect the answer to a randomly generated question with a random answer and give an accurate solution. Problems which computers can mostly solve using numerical input into formulas, are now able to be solved by the computer automatically detecting what a casual question is asking for, and what the proper solution is.

Recently OpenAI released an API for their AI GPT-3[18]. What it does is it uses data collected from the 2019 Internet via Web Scraping to determine the most probable answer to a given text input and can detect the most probable answer to a question using the info people have fed the internet back in 2019 when it was built. It is currently the most advanced and smartest AI on the market by far, representing a vast improvement over previous models such as GPT-2 and Microsoft Azure AI services [17]. It has over a billion parameters, which is dozens of times more than the number of parameters competitors like Microsoft and Google have.

Much recent research explored different capabilities of GPT-3. Brown et.al [7] studied the ability to solve SAT reading comprehension questions. Jun [4] researched how well GPT-3 can solve mathematical problems from project Euler and discovered that GPT-3 is very sensitive to the problem specification and performed better when it was requested to produce the python code that solves the problem instead of finding the problem numerical solution. Many similar studies were performed to evaluate the ability of AI models to solve trivia questions, math problems and even programming tasks from leetcode [8]

In this research, we further evaluated GPT-3 by testing its ability to answer mathematical word problems of varying types, one step and two step ones, with randomized names and objects. These types of problems can usually be found in middle-to-high school tests and even though arithmetic calculation is primitive, the text part requires certain reading comprehension, and ability to perform mathematical abstraction on different levels [9]. By testing GPT-3's ability to solve Algebraic problems we are trying to determine how advanced is the abstract knowledge of today's AI is, and how AI can use training datasets randomly generated or available on the internet to solve abstract problems which have been randomized in order to help test the AI's capability to use Human-like reasoning and logic.

In the absence of a convenient and available dataset we introduce a random word problem generator which is used to produce the problems that we then try to solve with the help of GPT-3. In order to make accurate evaluations we use statistical tests and try to find correlation between the accuracy of GPT-3 and input problem' parameters. By doing this we can try to build the problem set which can be used for retraining purposes thus improving GPT-3 accuracy.

AI has long been touted as the future, and how it will soon be able to solve most abstract problems. By testing GPT-3's ability to solve randomized word problems, we evaluate how well text models such as GPT-3 can solve typical problems met at high and middle school tests for students, such as PSAT and SAT. The typical arithmetic word problems is just a first step in this research that can potentially be adjusted to evaluate other types of tests. With this project, our overall goal is to introduce statistical instruments that evaluate the performance of neural networks and generate a set of problems intended for AI training purposes in order to improve accuracy of AI. The accuracy and precision of AI solving capabilities will help us to show how close we are to optimal performance, if there is any form of plateau in the quality that cannot be exceeded, and more generally, how close AI is to overtaking Human Beings in terms of abstract knowledge and general intelligence.

METHODS

DATA COLLECTION

Due to the lack of a suitable dataset in the literature that tested algebraic tasks across a diversity of problems, we decided to generate problems ourselves using a program generator written by ourselves. The primary reason we used this approach was the absence of uniformity in format of problems and low number of test cases. We first created one step problems, then customized our functions to be able to create one or two step problems. We basically created functions which can create 8 types of problems, addition, subtraction, multiplication, and division problems with one or two steps. We also made sure to include a massive database of objects, names, and numbers to properly randomize problems and verified that the result is a natural number.

Below are few examples of the generated problems:

- *Scott has 1 Banana, then he multiplies the amount of Bananas he has by 3, afterwards Scott adds 87 more Bananas. How many Bananas does Scott have?*
- *Jay has 4 phones, then Ja 23 more phones, afterwards he divides the amount of phones by 9, how many phones does Jay have now?*

DATA ORGANIZATION

Each test run consists of a selected set of problems given to GPT-3 as input. Problems in the test set are randomly generated to prevent the data from being skewed. For every test run we record the following statistical parameters of the problems from the test set:

- average number size,
- problem length,
- euler distance (see details below)
- Other statistical parameters

The graphical user interface for problem generator (programmed in IPython) allows users to produce a csv file with all of the word problems inside of it as an output . Users can choose how many problems should be included in the output dataset and their parameters. We show multiple statistical tests on the users screen, using the data generated by the functions we made.

INTERACTION WITH GPT-3

In order to connect to GPT-3 we used the OpenAI interface [10, Sumarak].We used the Davinci Model which has 175 billion of trainable parameters[11,Khan], and is considered to be one of the most advanced GPT-3 implementations targeted to solve logic problems. We were able to feed the problems to the AI and get proper responses which were not too far from the actual answers we also generated. We introduced token randomization and waiting timeout to help overcome limitations on the number of API requests per second to avoid being rate limited by the OpenAI module. We recorded all the properties of each problem in a dictionary, which we updated every time the program ran. This data is used as the foundation for the conducted statistical tests, and to get the results which are detailed enough to generalize a conclusion about GPT-3's ability to solve word problems.

ANALYSIS OF GPT-3 PERFORMANCE

The typical output of GPT-3 to the world problem is a sentence that requires parsing:(*{Name} has {answer} {Objects} Left.*). Hence, in order to perform quantitative analysis of how well GPT-3 answers the question we had to convert the output of GPT-3 to some normalized form. The reason why we don't compare the results as numbers is because GPT-3 is text processor, thus it doesn't make sense to penalize, for example, difference between 90(right answer) and 901(GPT-3 answer) more than the difference between 90 and 91: overall, this is just 1 symbol mistake in both cases. We encoded all of the strings into a vector representing the contents of the string. Initially we tried using the OpenAi encoder, but it was slow and would also send a lot of API requests. As a result, we used the BERT model [12] which uses an encoder that runs locally on the computer, and does not send any requests to the OpenAi website.

We then determined how accurate the response of GPT-3 is by comparing the similarity between the response vector and the generated answer vector. We found out that the Cosine Similarity formula, which involves the cosine of two vectors and returns a single similarity constant, works very well.

$$similarity = cos(\boldsymbol{\theta}) = \frac{A+B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 1 Cosine Similarity Formula

As an alternative we considered the vector distance formula. Vector distance was not very efficient as it would often result in deceiving distances from two answers, and overall Cosine similarity functioned much better than vector distances. Cosine similarity is used in most AI Recognition software involving human input via text and strings, and is used to determine the weight and overall, the most probable answer to a question. Hence it is used by most AI researchers, as it is probably one of the most efficient formulas for determining differences between two sentences[15]. We made sure to use similar size vectors so the results would be the best and most accurate. We used this formula to record the distance between two encoded strings, which are essentially vectors, and put them in the main dictionary. We did this for all of the problems, making sure that we had properly formatted data.

We finally ran multiple statistical tests, which helped confirm the results about how well the GPT-3 AI solves certain problem types. We used the statistical T-test which is a common statistical test used to determine the similarity between two statistics.

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{S_p^2\left(\frac{1}{n1} + \frac{1}{n_2}\right)}}$$

Figure 2 T-Test

We then analyzed the resulting p values, using a cutoff of $p < .05$ to determine if our results were significant or not. We often found out that our values would have a varying distance, and we had to include other factors in our graphs to make sure that our graphs are as descriptive as possible. We used the statsmodel module [13] in order to get graphical representation of inferred results from the data we've got.

We then would launch our program multiple times to help create a large model. In the end, we use the converged data from multiple runs, and apply statistical analysis on top in order to confirm our theories and hypotheses.

RESULTS

The following results were obtained after performing statistical analysis. First of all, we found that generally simple addition and multiplication problems with only one step were mostly answered correctly. This is very reasonable since GPT-3 is trained on answering straightforward and non-complex questions. As for two-step problems, GPT-3 performance was quite disappointing which can be seen in the results that we got below, and they help confirm the hypothesis we made about AI to some extent, and how it's not quite ready to solve complex abstract problems.

The distance between the resulting vectors is much larger for problems with two steps, and problems which require more complex operations like multiplication and division.

We then tried to create a new attribute called 'AverageNumSize', which is the average number size of all of the numbers in the problem. What we got was very interesting. The scatter plot has lots of variation, but we can see that as the numbers get larger, the distances between the correct answer and AI's answer converge into a small range. There is no definitive proof that this pattern really means anything, but we suspect that it could be explained by the fact that the "price" of distance error is less for bigger numbers but the probability of error increases with growing numbers.

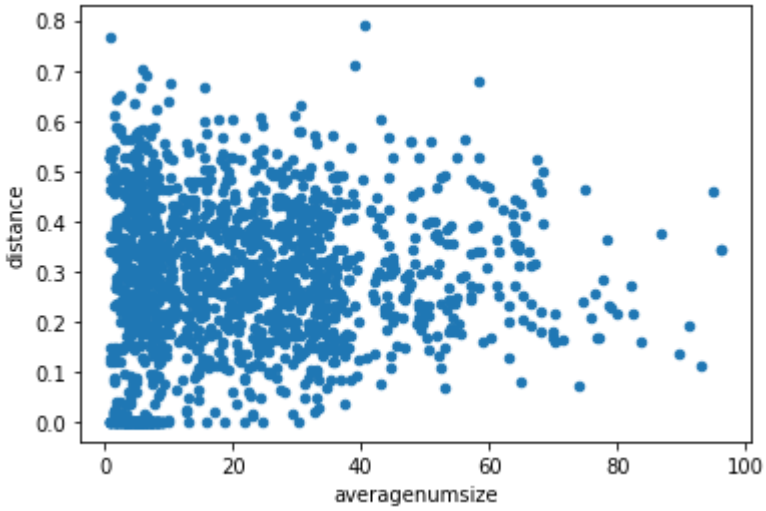

Figure 3 Scatter Plot of the average of the numbers in each problem

The unexpected result was that multiplication problems, which require two steps of multiplication, had nearly the same average distance as problems with one division step. That is somewhat counterintuitive since it is natural to assume that a more complex problem would result in a significant difference, which is contradicted by the relatively small difference between division and two step multiplication.

Another interesting thing is that addition is almost always easily solved by GPT, but it regularly has problems with subtraction. Reasonably, due to GPT-3's nature, we would think that due to the fact that it was trained by data collected by web-scraping the internet, they would have similar correct response rates, yet their rates are completely different. Also, by looking at the scatter plot, we can see that this change can't be explained by simple variation in numbers, since the number size does not matter. This again shows me that the previous hypothesis, which was that larger number problems are harder to solve, is mostly wrong.

One pattern we do see is that in general problems with addition and multiplication are on average easier for GPT-3 AI to solve. This is very significant for our future research and is completely accepted as a null hypothesis by statistical tests through the test data. Hence, we can come to a completely new hypothesis, which is that perhaps complication is not the only factor which affects the average number of problems the AI gets right, but perhaps the

type also affects the number of problems the AI gets right. We tried reformatting the problems and GPT-3 answer pattern multiple times, but in general the pattern remained the same, where on average addition and multiplication problems were more easily solved than division and subtraction.

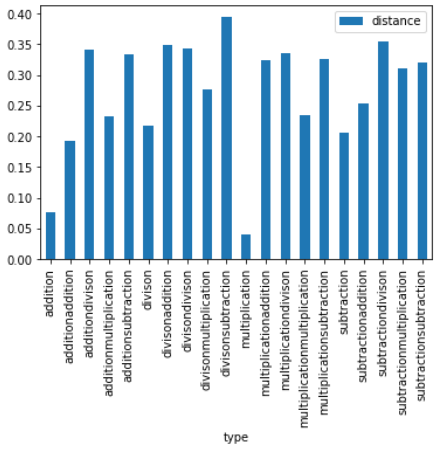

Figure 4. Results by operation

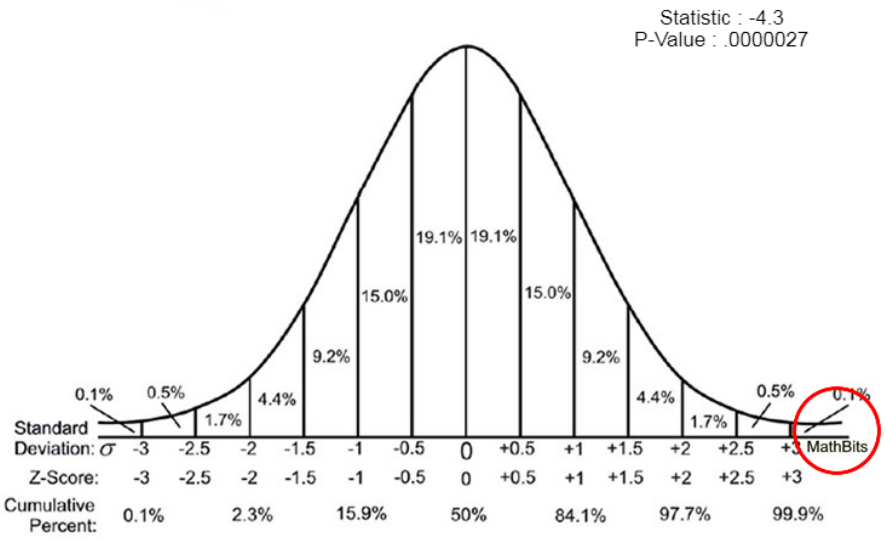

Figure 5. T-Test Visualization

CONCLUSION AND FUTURE WORK

Currently GPT-3 represents how the current AI is not able to solve many abstract and randomized word problems that require multiple steps. Modern AI is simply not able to grasp many abstract human concepts. Although this is currently true, AI is becoming increasingly intelligent. As AI becomes more able to solve abstract problems, we can see that the current difference in solvability between types of problems might even disappear, and eventually AIs like the GPT-3 AI will be able to solve problems which have two or more steps.

To further test the mathematical capabilities of GPT-3, we plan to continue research in several directions. First and most important of all, we have used a pretrained model implementation of GPT-3 without the ability to improve quality of answer by providing additional data sets for training. We believe that additional training based on

a set of mathematical problems might potentially improve the results. Second, a research of using GPT-3 for project Euler problems [4] showed that if GPT-3 is asked to provide the result in a programmatic way (i.e. we basically ask to find the solution as a program) it significantly improves GPT-3 performance. So we would like to try this approach as well and compare results. Finally, we would like to start experiments with more complex mathematical problems with the eventual goal of finding a way to make GPT-3 solve SAT-level mathematical problems.

To summarize: the current limitations are that GPT-3 is most likely trained on a relatively limited set of mathematical data which limits its capabilities to do math. The internet is so diverse that mathematical content only makes up a minor part of it. Even so, GPT-3's ability to solve randomized problems is very impressive, nonetheless. As new AI's continue to be developed, we can see that there will soon be AI's which would be able to grasp mathematical concepts, not from a list of data like GPT-3, but through actual intelligence and cognitive abilities. We may soon see an AI which would be able to solve all word problems efficiently and without mistakes.

ACKNOWLEDGEMENTS

I would like to thank my personal advisor, Tyler Giallanza, for helping me with this whole project, invigorating my interest in scientific research and guiding me through introduction to Python programming and GPT-3 study. I would also like to thank my teachers at Castro Valley High School.

REFERENCES

1. "Ai's Golden Age." *Artificial Intelligence*, UBS, https://www.ubs.com/microsites/artificial-intelligence/en/golden-age.html.

2. Sagar, Ram. "What's next for AI: Solving Advanced Math Problems." *Analytics India Magazine*, Analytics India Mag, 15 Feb. 2021, https://analyticsindiamag.com/whats-next-for-ai-solving-advanced-math-problems/.

3. Freeland, Devon. "Modern Algebra." *How Has Algebra's Methods Changed Over Time?*, https://algebrasmethods.weebly.com/modern-algebra.html#:~:text=Algebra%20has%20really%20affected%20the%20human%20race%20more,huge%20scale%20to%20create%20extremely%20complex%20computer%20system.

4. *Prompt Engineering GPT-3 to Solve Project Euler Problems*. https://towardsdatascience.com/prompt-engineering-gpt-3-to-solve-project-euler-problems-1ff3b12f7d56.

5: Khanam, Sana, et al. "Artificial Intelligence Surpassing Human Intelligence: Factual or Hoax." *OUP Academic*, Oxford University Press, 2 Jan. 2020, https://academic.oup.com/comjnl/article/64/12/1832/5688168.

6. Gkionaki, Melina. "How Does Artificial Intelligence Work?" *European Investment Bank*, European Investment Bank, 9 Feb. 2022,

7. Brown, et .al "Language Models are Few-Shot Learners"

8. "Competitive Programming with AlphaCode." *RSS*, https://www.deepmind.com/blog/competitive-programming-with-alphacode.

9. Kathryn Rich, Aman Yadav "Applying Levels of Abstraction to Mathematics Word Problems"

10. Sumrak, Jesse. “What Is GPT-3: How It Works and Why You Should Care.” *Twilio Blog*, Twilio, 27 July 2021, https://www.twilio.com/blog/what-is-gpt-3.

11. “What Is GPT-3 and Why Is It Important?” *RSS*, https://www.genei.io/blog/what-is-gpt-3-and-why-is-it-important?genei_segment_id=327ae72a-198b-4acc-85d7-133b7ca31445.
12. https://pypi.org/project/bert/

13. https://pypi.org/project/statsmodels/

14. *DeepMath - Deep Sequence Models for Premise Selection*. https://arxiv.org/pdf/1606.04442.pdf.

15. Prabhakaran, S. (2022) *Cosine similarity - understanding the math and how it works? (with python)*, *Machine Learning Plus*. Available at: https://www.machinelearningplus.com/nlp/cosine-similarity/ (Accessed: November 12, 2022).

16. Možina, Martin, et al. “Argument Based Machine Learning.” *Artificial Intelligence*, Elsevier, 29 Apr. 2007, https://www.sciencedirect.com/science/article/pii/S0004370207000690.

17. “A Comparison of GPT-3 and Existing Conversational AI Solutions.” *HackerNoon*, https://hackernoon.com/a-comparison-of-gpt-3-and-existing-conversational-ai-solutions-0q2z3z9x.

18. Heaven, Will Douglas. “OpenAI's New Language Generator GPT-3 Is Shockingly Good-and Completely Mindless.” *MIT Technology Review*, MIT Technology Review, 20 Oct. 2021, https://www.technologyreview.com/2020/07/20/1005454/openai-machine-learning-language-generator-gpt-3-nlp/.