Training and Evaluating Machine Learning Transformer Models to Perform Various Mathematical Tasks

Eric Hury, Citrus College
Dr. Yu Bai, Department of Computer Engineering, California State University, Fullerton

Background

Transformer based Neural Networks are the basis for many current generation artificial intelligence and generative text-based chatbots. Open AI's ChatGPT, Google's BARD and Microsoft's Bing AI chatbot all use transformer based neural networks to take a text-based input and generate conversational text-based outputs.

This project seeks to understand how a transformer based neural network, developed and trained with language and speech, will behave when instead presented and trained with a mathematical operation.

Datasets were sourced from work by Saxton, D. et al.[1], Piotrowski, B. et al.[2], and Lample G. et al.[3]. Code and data provided by Bartosz Piotrowski.[4]

Hypothesis

Since transformer based LLMs (Large Language Models) are technologically capable of human speech translation and conversation imitation, we predict that the same technology will be able to "translate" a given mathematical input into it's solved or simplified form.

Given two training conditions, with models both pretrained and untrained on language data, we predict that the pretrained model will be able to translate a given mathematical operation more accurately.

Methods

- Data Collection:
  The training and testing data consisted of mathematical equations from curated datasets from Bartosz Piotrowski.

- Data Preprocessing:
  The raw mathematical equations were preprocessed to convert them into a suitable format for training the transformer models.

- Model Selection:
  Two transformer models, T5 (Text to Text Transfer Transformer) and GPT2 (Generative Pre-trained Transformer 2), were chosen for the task of equation solving.
  GPT-2 excels in language generation and completion, while T5 is designed for a wide range of NLP tasks by framing them in a consistent text-to-text format.

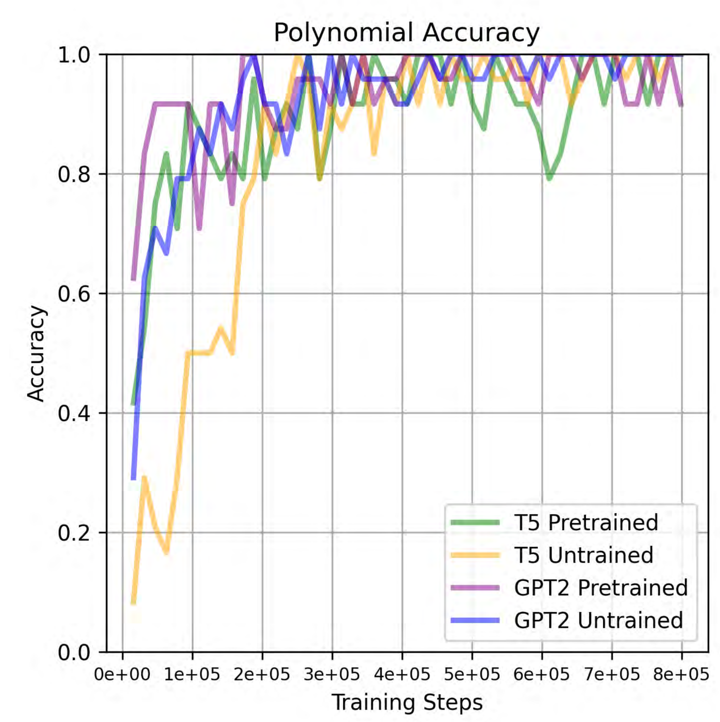
- Model Architecture:
  The T5 model and GPT2 model were adapted to the equation solving task using PyTorch, a popular deep learning framework.

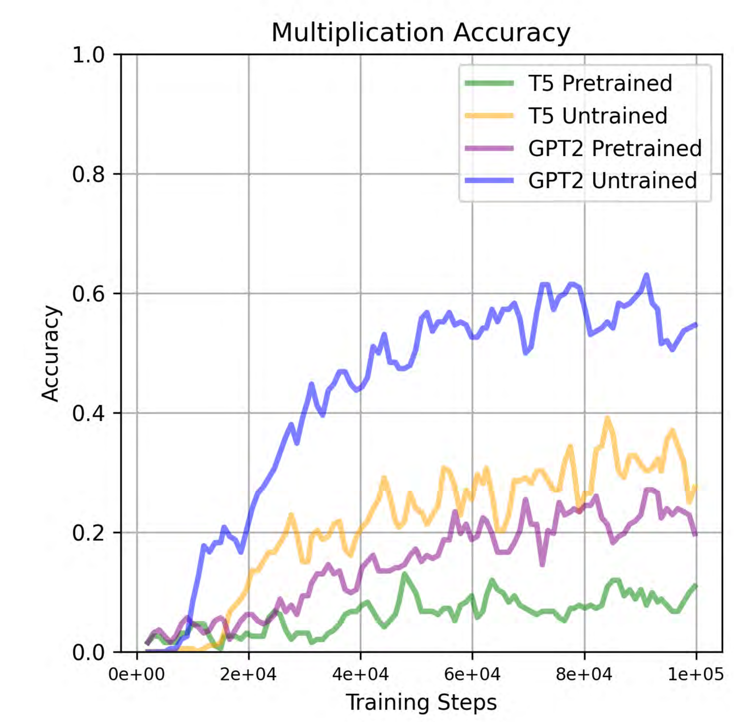
- Model Training:
  The fine-tuning process involved training the models on the equation dataset using gradient descent optimization with PyTorch.
  The models were trained for multiple epochs with various batch sizes.

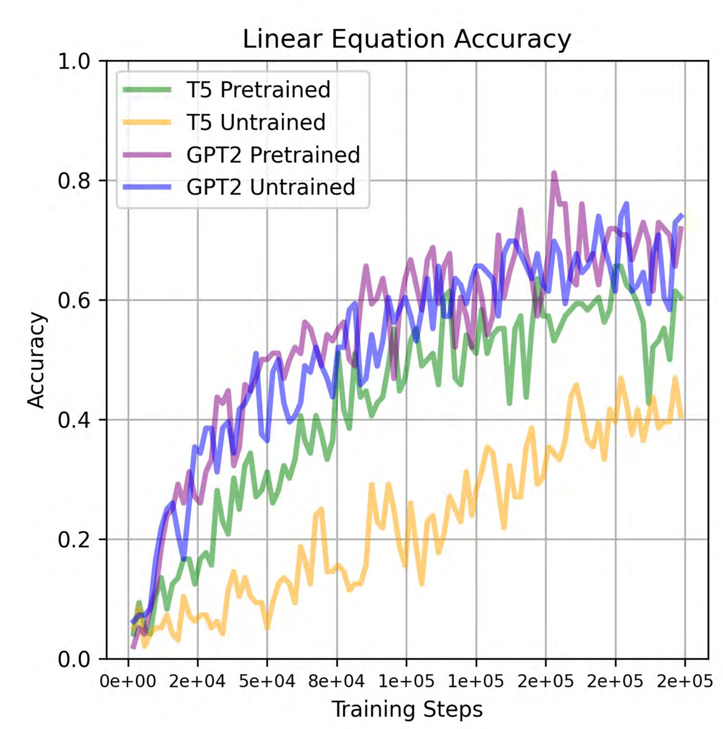
- Model Evaluation:
  The trained T5 and GPT2 models were evaluated on a separate test set consisting of unseen equations to verify model accuracy.

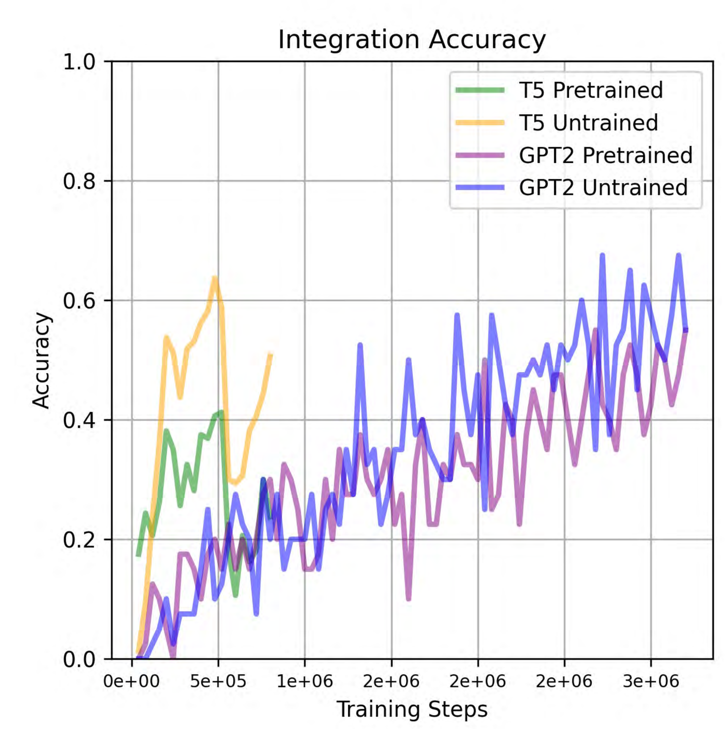
- Hardware and Software:
  The experiments were conducted on a machine equipped with CUDA-capable NVIDIA GPUs for efficient training of the transformer models.
  The implementation was done using Python programming language and PyTorch for model training and evaluation.

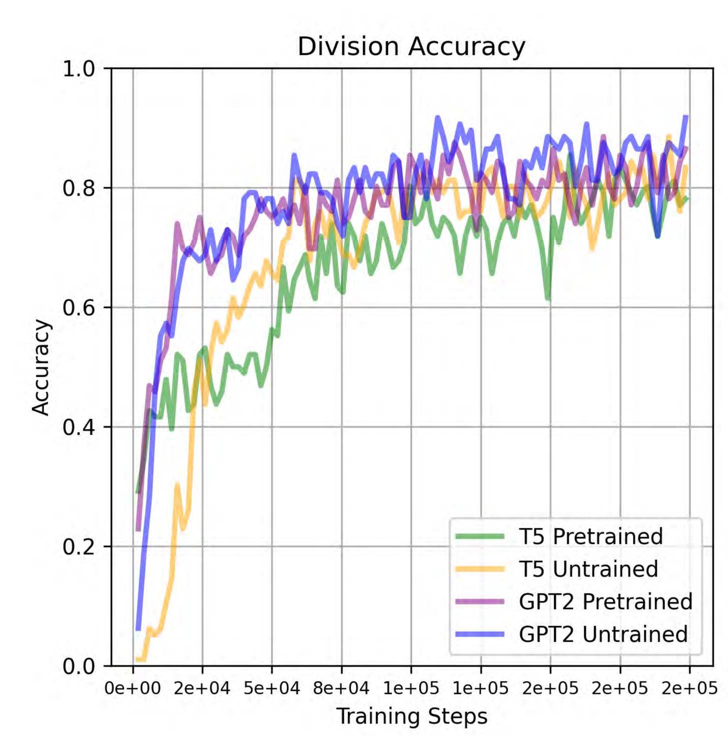
Results

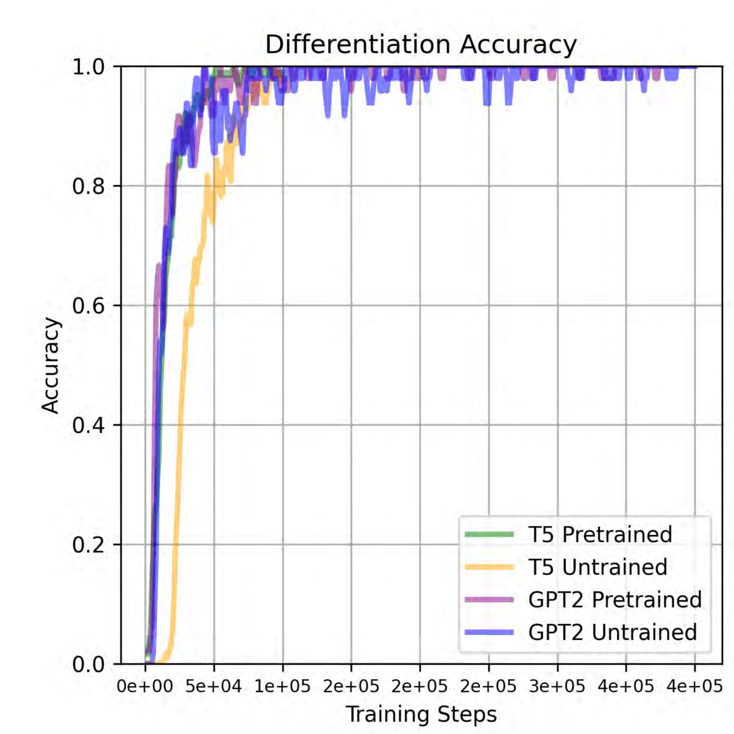


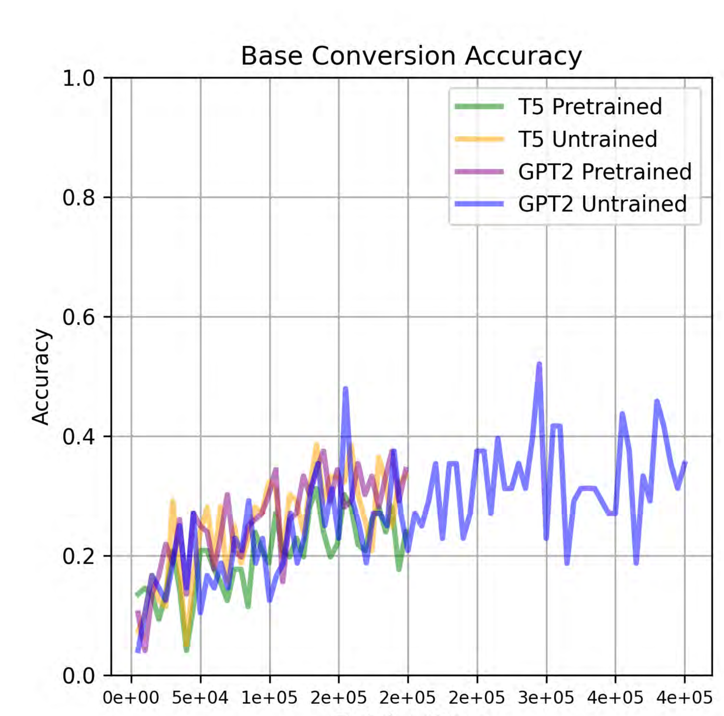












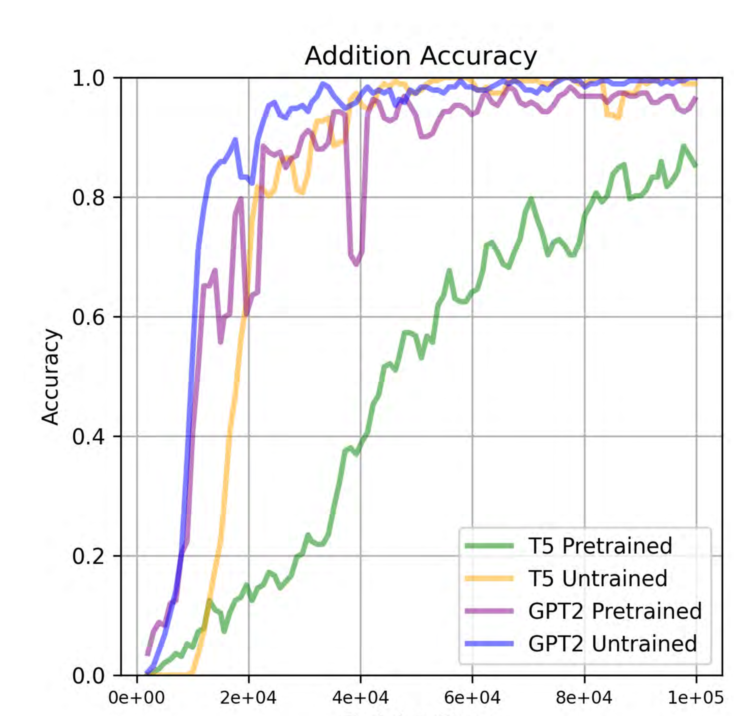


Table 1: Final accuracy of neural language models trained and tested on eight datasets.

Dataset	T5		GPT2	
	Untrained	Pretrained	Untrained	Pretrained
Polynomial Normalization	91.33%	58.21%	93.03%	90.15%
Multiplication	45.18%	23.42%	65.98%	38.91%
Linear Equations	17.70%	30.99%	53.88%	54.01%
Integration	35.60%	21.63%	62.83%	54.29%
Division	71.85%	67.73%	78.72%	73.81%
Differentiation	95.90%	98.95%	99.44%	95.39%
Number Base Conversion	0.61%	0.12%	5.65%	3.08%
Addition	97.54%	83.30%	98.32%	96.03%

Conclusion

Using the given datasets, we were able to determine the accuracy of LLMs trained with various mathematical operations. The untrained models were more accurate overall apart from linear equations via both T5 and GPT2 as well as Differentiation via T5. This suggests that pretrained data doesn't include any information that helps with mathematical operations. It's also worth noting that GPT2 preformed better in a majority of the datasets, with the subtle exception of pretrained differentiation via T5.

The results of this research are promising for intuitive mathematical reasoning. While the precision of LLMs for mathematics is unacceptable for most mathematical use cases, machine-lead intuitive reasoning may lead human researchers to otherwise unforeseen solutions.

Future Work

Analyzing accuracy over time, additional training steps in several models could have led to an increase in model accuracy.
Additionally, the field off LLMs and Machine Learning has greatly improved since T5 and GPT2 were released. It would be interesting to test these mathematical operations on other, more current NLP's and machine learning models.

References

[1] Saxton, D., Grefenstette, E., Hill, F., and Kohli, P. Analysing math- ematical reasoning abilities of neural models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019* (2019), OpenReview.net.

[2] Piotrowski, B., Urban, J., Brown, C. E., and Kaliszyk, C. *Can neural networks learn symbolic rewriting?* CoRR abs/1911.04873 (2019).

[3] Lample, G., and Charton, F. Deep learning for symbolic mathematics. *In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (2020), OpenReview.net.

[4] Piotrowski, B. (2022). transformers-for-mathematics. GitHub. https://github.com/BartoszPiotrowski/transformers-for-mathematics

Acknowledgements

Project supported by Project RAISER, U.S. Department of Education HSI-STEM award number P031C210118.

Project supported by Project RAISE, U.S. Department of Education HSI-STEM award number P031C160152.

Alternate Text

Eric Hury

Cal State Fullerton

'Training and Evaluating Machine Learning Transformer Models to Perform Various Mathematical Tasks'

Background: Transformer based Neural Networks are the basis for many current generations' artificial intelligence and generative text-based chatbots. Open AI's ChatGPT, Google's BARD and Microsoft's Bing AI chatbot all use transformer based neural networks to take a text-based input and generate conversational text-based outputs. This project seeks to understand how a transformer based neural network, developed and trained with language and speech, will behave when instead presented and trained with a mathematical operation. Datasets were sourced from work by Saxton, D. et al.1, Piotrowski, B. et al.2, and Lample G. et al.3. Code and data provided by Bartosz Piotrowski.4

Hypothesis: Since transformer based LLMs (Large Language Models) are technologically capable of human speech translation and conversation imitation, we predict that the same technology will be able to "translate" a given mathematical input into its solved or simplified form. Given two training conditions, with models both pretrained and untrained on language data, we predict that the pretrained model will be able to translate a given mathematical operation more accurately.

Methods:
•*Data Collection:*
The training and testing data consisted of mathematical equations from curated datasets from Bartosz Piotrowski.
• *Data Preprocessing:*
The raw mathematical equations were preprocessed to convert them into a suitable format for training the transformer models.
• *Model Selection:*
Two transformer models, T5 (Text to Text Transfer Transformer) and GPT2 (Generative Pre-trained Transformer 2), were chosen for the task of equation solving.
GPT-2 excels in language generation and completion, while T5 is designed for a wide range of NLP tasks by framing them in a consistent text-to-text format.
• *Model Architecture:*
The T5 model and GPT2 model were adapted to the equation solving task using PyTorch, a popular deep learning framework.
• *Model Training:*
The fine-tuning process involved training the models on the equation dataset using gradient descent optimization with PyTorch.
The models were trained for multiple epochs with various batch sizes.
• *Model Evaluation:*
The trained T5 and GPT2 models were evaluated on a separate test set consisting of unseen equations to verify model accuracy.
• *Hardware and Software:*
The experiments were conducted on a machine equipped with CUDA-capable NVIDIA GPUs for efficient training of the transformer models.
The implementation was done using Python programming language and

PyTorch for model training and evaluation.

Results:
Numerous Charts titled: *'Polynomial Accuracy', 'Multiplication Accuracy', 'Linear Equation Accuracy', 'Integration Accuracy', 'Division Accuracy', 'Differentiation Accuracy', 'Base Conversion Accuracy', and 'Addition Accuracy'.*
Table 1: Final accuracy of neural language models trained and tested on eight datasets.

Conclusion: Using the given datasets, we were able to determine the accuracy of LLMs trained with various mathematical operations. The untrained models were more accurate overall apart from linear equations via both T5 and GPT2 as well as Differentiation via T5. This suggests that pretrained data doesn't include any information that helps with mathematical operations. It's also worth noting that GPT2 performed better in a majority of the datasets, with the subtle exception of pretrained differentiation via T5. The results of this research are promising for intuitive mathematical reasoning. While the precision of LLMs for mathematics is unacceptable for most mathematical use cases, machine-lead intuitive reasoning may lead human researchers to otherwise unforeseen solutions.

Future Work: Analyzing accuracy over time, additional training steps in several models could have led to an increase in model accuracy. Additionally, the field off LLMs and Machine Learning has greatly improved since T5 and GPT2 were released. It would be interesting to test these mathematical operations on other, more current NLP's and machine learning models.

References:
[1] Saxton, D., Grefenstette, E., Hill, F., and Kohli, P. Analysing math- ematical reasoning abilities of neural models. In 7^{th} *International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019* (2019), OpenReview.net.
[2] Piotrowski, B., Urban, J., Brown, C. E., and Kaliszyk, C. *Can neural networks learn symbolic rewriting?* CoRR abs/1911.04873 (2019).
[3] Lample, G., and Charton, F. Deep learning for symbolic mathematics. *In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (2020), OpenReview.net.
[4] Piotrowski, B. (2022). transformers-for-mathematics. GitHub.
https://github.com/BartoszPiotrowski/transformers-formathematics

Acknowledgements: Project supported by Project RAISER, U.S. Department of Education HSI-STEM award number P031C210118.
Project supported by Project RAISE, U.S. Department of Education HSI-STEM award number P031C160152.