Review OF GPT-3, How it outperforms other AI models

Geeta Shivani*, Deepika Dash**
*(Computer Science, Rashtreeya Vidyalaya College of Engineering, Bengaluru
geetashivanit.cs18@rvce.edu.in)
** (Computer Science, Rashtreeya Vidyalaya College of Engineering, Bengaluru
deepikadash@rvce.edu.in)

--------------------------------------***************************---------------------------------

Abstract:

A neural network machine learning model trained using internet data called GPT-3, or the third generation Generative Pre-trained Transformer, can produce any kind of text. It was created by OpenAI, and it only needs a tiny quantity of text as an input to produce huge amounts of accurate and complex machine-generated text. A ground-breaking work titled Language Models Are Few-Shot Learners was released by Open AI in May 2020. They demonstrated GPT-3, a language model that boasts 175 billion parameters and holds the record for being the largest neural network ever built. Consequently, it is 100 times bigger than the second generation (GPT-2). Based on the tests that GPT-3 is capable of completing, we may see it as a model that is capable of completing reading comprehension and writing tasks at a level that is nearly human, with the exception that it has read more text than any human will ever read in their lifetime. Consequently, is a very potential model.

Keywords — – **GPT-3, csv files, OenAI Playground. Pytorch, Gopher, Chinchilla**

--------------------------------------***************************---------------------------------

I. INTRODUCTION

The training data includes both natural language and billions of lines of code from publicly accessible sources, such as code in open GitHub repositories. OpenAI Codex is a successor of GPT-3. Python is the language in which OpenAI Codex excels, but it also knows how to use Javascript, Go, Perl, PHP, Ruby, Swift, Typescript, and even shell. In comparison to GPT-3, which only has 4KB of memory for Python code, it has 14KB, allowing it to process over 3 times as much contextual data.

The primary ability of GPT-3 is to produce natural language in response to a natural language prompt, therefore the only way it can change the world is in the reader's imagination. Even while OpenAI Codex creates functioning code, it has a large portion of GPT-3's natural language understanding, allowing you to give any piece of software having an API command in English. As a result of OpenAI Codex, everyone will be able to use computers more effectively since they will be able to better understand people's intentions.

With the aid of Codex, OpenAI shows how the program can be used to create basic games and websites using natural language, as well as translate between other programming languages and respond to data science queries.GPT-3 has gained its popularity for text production because it outperforms other AI models in its capacity to generate poetry, play chess, perform arithmetic, and write web interface code based on requirements expressed in natural language.

Kaplan initially demonstrated that the parameter for improving performance is an increase in model size. This hypothesis gave rise to OpenAI, which is superior to GPT-2, BERT, and chinchilla. A few additional AI services, like LaMDA, Jurassic-1, and Gopher, are based on the same power law as openAI and recommend that, the more money available for model training, the majority of it should be used to increase the size of the models. Kaplan's finding was examined by deep brain researchers, who found that data, in addition to model size, is a crucial component for improving performance. A smaller model can easily outperform a larger model because, for example, when we double the size of the model, the number of training tokens should also be increased. Gopher, another LLM created by Deep Mind, performs uniformly and much better than Gopher and other LLMs despite having been trained on four times as much data as Chinchilla.

Many of us have always found it difficult to code, therefore here is a solution and a technological improvement. If we pose an inquiry in this case, we will receive a command in return. However, there are two issues: the first is token exhaustion, and the second is accuracy. Therefore, my objective for the project was to develop a programme that accomplishes the same task as OpenAI Playground while simultaneously resolving the aforementioned problems.

II. RELATED WORK

We will now review various methods for transforming text into commands.

Then, in February 2019, OpenAI unveiled GPT-2 (for Generative Pre-trained Transformer 2), a sizable unsupervised transformer language model trained on 40GB of text, or around 10 billion tokens, with 1.5 billion parameters. The model was able to generate very coherent and convincing-sounding output when used to forecast the next word in a text based on the prior context, but it could also generate gibberish. Instead of making the model itself available, access was to be made available via an API, providing the model's developers more control over how it was used. As of this writing, a beta version of the API is available, but if you want access, you'll need to sign up for the probably lengthy wait list.

In particular, the researchers identified tasks where increased model scale led to improved accuracy, such as reading comprehension and fact-checking, as well as those where it did not, such as logical and mathematical reasoning. The team assessed Gopher using a variety of NLP benchmarks, such as Massive Multitask Language Understanding (MMLU) and BIG-bench, and compared its performance to several baseline models, including GPT-3. They found that Gopher consistently improved at knowledge-intensive tasks but not as much at tasks requiring strong reasoning. Language models are referred to as autoregressive when they are applied iteratively with the anticipated outcome supplied back as the input.

Both LaMDA and GPT-3 are trained on unlabeled text datasets. For instance, Wikipedia and Common Crawl were used in the training of GPT-3. Using dialogue training sets, LaMDA is taught to generate non-generic, open-ended dialogue that is truthful, rational, and relevant to the problem. For instance, during a presentation at Google I/O 2021, LaMDA presented data on Pluto, space travel, and user comments in a style that resembles human voice and emotions. Access was to be made available via an API rather than the model itself, giving model makers more control over how it was utilised. A beta version of the API is accessible as of the time of writing, but if you want access, you must join the doubtless long wait list.

Some information about future pricing has inadvertently leaked through a typical SaaS tiered pricing model, which includes a free tier that grants access to 100,000 generated tokens, a US$100 per month tier that grants access to 2 million tokens, and a US$400 per month tier that grants access to 10 million.

These prices have drawn criticism for being excessively costly; they unquestionably exceed the normal price threshold of under $50 per month for many other SaaS providers.

III. PROBLEM STATEMENT

Technology, especially advancements in AI, fascinate us because of how quickly our environment is changing. With a 78.1 percent accuracy in the one-shot situation and a 79.3 percent accuracy in the few-shot scenario, GPT-3 exceeds a tuned 1.5B parameter language model and accuracy of 75.4 percent. The prompt must be written in such a way that it gives us the required code, and that is our main objective. It's possible that additional models will be employed as well, but testing them will take time because they're still in the beta stage.

IV. METHODOLOGY

A. *Evaluation of Accuracy*

It has been examined using OpenAI. In this project, the gpt3 computer has been given a sentence in everyday English, and it is expected to respond with the proper command based on the input. On a sizable dataset, a later training and testing step is being conducted. Before gradually expanding the dataset size, the testing should start with a smaller dataset so that errors or false positives can be quickly identified.

B. *Input generation*

Now in order to give input the file should be of jsonl format. To achieve this format a json file needs to be converted to jsonl. The purpose of using jsonl over json is that it is a convenient format for storing structured data which may process one record at a time. For doing so we use google collab otherwise OpenAI playground could be used.

C. *Prediction*

If the prompt design isn't defined properly, we cannot get the expected outcome, so it is very necessary to design the prompt efficiently in order to get the appropriate result.

D. *Pytorch*

Since GPT-3 OpenAI has token limit we have executed it in pytorch in order to design the model as per our convenience and our requirement.

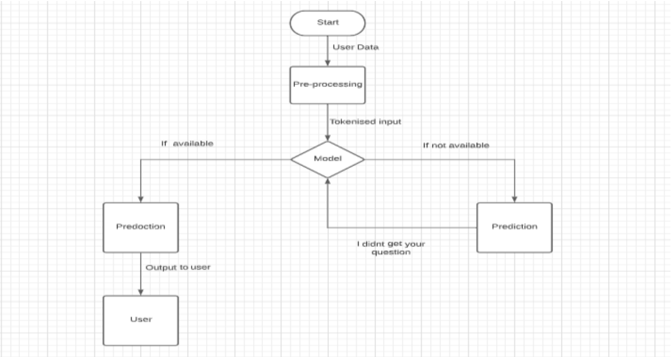

Fig.1 Flowchart of the OpenAI Model

V. GPT-3 ARCHITECHTURE

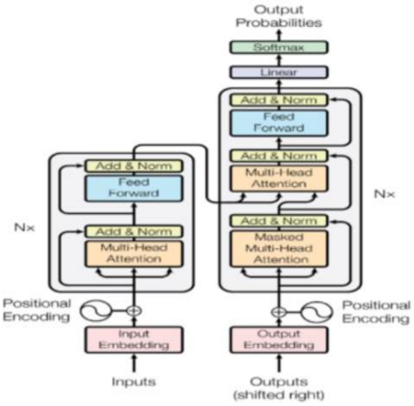

Fig. 1 GPT-3 Architecture

As we can see there are two modules one that is left one which is called encoder module and right one is the decoder module. Further in this chapter I have given the detailed explaination of the architechture module wise. Starting firstly ewiththe encoder module. This is our first step in encoder module where the given input is converted to word which is called as word embedding. Word embedding is a lookup table which converts any particular word gets converted to representation of a number then to vector with continuous values. Next step is to provide positional information to embedding. This process is called positional embedding. From the below figure where we can see the formule.Here, sine function is used for even index and cos function for odd index position.

Encoded layer sub module’s function is to map all the input sequence into abstract continuous representation which holds learned information of the entire sequence. This multi headed attention can be achieved by encoded self-attention mechanism. The self-attention model allows inputs to interact with each other (i.e. compute attention of all other inputs with respect to one input), whereas the attention mechanism allows output to focus attention on input while creating output.

By locating the attention score matrix, the self-attention operation is encoded. The value matrix may pay greater attention to significant words while avoiding irrelevant words if the attention score matrix is used as a filtering matrix. The attention score matrix can be obtained by multiplying the question and key matrices. The scores are then scaled to enable more stable gradients. Scaled scores make it possible to compare results accurately and prevent unfair advantages for those who took the easier test from being given to those who took the more difficult test. Next we have to find softmax of a matrix. Then the attention weights are multiplied by value in order t get the output. After which we let them pass through linear function to get the final result. In neural network models that predict a multinomial probability distribution, the softmax function is used as the activation function in the output layer. In other words, softmax is used as the activation function in multi-class classification problems where class membership on more than two class labels is required.

Residual connection layer allows gradient to flow through network directly. Residual connections are synonymous with skip connections. They allow gradients to flow directly through a network without passing through non-linear activation functions.In normalization layer is used to stabilize the process. The inclusion of normalisation layers in the model frequently aids in the acceleration and stabilisation of the learning process. Batch Normalization could be used if training with large batches is not an issue and the network does not have any recurring connections.Pointwise feed forward **i**s used to project the attention outputs. The pointwise feed-forward network consists of two linear layers separated by a ReLU activation. The output of that is then normalised and added to the input of the pointwise feed-forward network. The residual connection between the point-wise feedforward layer's input and output.

In decoder layer output embedding and positional encoding is done in a similar way as is done in encoder layer. From there masked scores are sent to multi headed attention. First multi headed attention layer is used to add look ahead mask scores to scaled scores. This process is called as masking. Here we need to SoftMax the masked scores. This is used to leave 0 attention score for future tokens. That is attention is only given to important words first and least important are just left behind. There are two multi headed attention layers used in decoder layer as both the process are required in order to get into conclusion. This layer’s main function is to match the inputs of encoder and decoder.

VI. CONCLUSIONS

GPT-3 has become well-known for text generation since it outperforms other AI models. On the basis of needs specified in natural language, the model can compose poetry, play chess, do math, and develop web interface code, in addition to the technology's successes in a wide range of other fields, some of which are fairly surprising.

The OpenAI API can be used for almost any task that requires understanding or generation of natural language or code. We provide a range of models with varying levels of power suitable for various tasks, as well as the ability to fine-tune your own custom models. These models can be used for a variety of tasks, including content generation, semantic search, and classification.

VII. FUTURE ENHANCEMENTS

GPT-3 was the largest neural network ever created at the time. OpenAI — and the rest of the world, given how quickly others followed their lead — realised that large pre-trained language models were AI's answer to the mysteries of human language.

GPT-3, on the other hand, possessed abilities that not even OpenAI's researchers had considered. Sharif Shameem was among the first to notice GPT-3's coding abilities. He successfully persuaded the system to create a generator that wrote code for him. The world was on the verge of one of the most significant AI revolutions in history.GPT-3 Codex appears to be impressive. Not because it can code in multiple languages or because it does it well. But because its interpretations of the English prompts are deep, nuanced, creative, and precise — despite the fact that it does not understand language in the same way that we do.

Not only this we can create a game using GPT-3 and talking to the computer. It just feels like you are talking to anyone just like your friend. The chatbot is so well programmed and has a wide range of pre trained models has a vast knowledge. Therefore, when we use OpenAI playground it is a wonderful yet seamless experience one should have. In the future more content and accuracy will be improved. In my project I have as an alternative to OpenAI where tokens would get exhausted is creating an app which functions similar to it. But the scope is limited as I have written the code for what I wanted to test, so therefore in the coming future I would like to increase the dataset used and ways to improve their accuracy.

REFERENCES

[1] Dale, Robert. "GPT-3: What's it good for?." *Natural Language Engineering* 27.1(2021):113-118.

[2] Floridi, Luciano, and Massimo Chiriatti. "GPT-3: Its nature, scope, limits, and consequences." *Minds and Machines* 30.4 (2020): 681-694.

[3] Thiergart, Jonas, Stefan Huber, and Thomas Übellacker. "Understanding Emails and Drafting Responses--An Approach Using GPT-3." *arXiv preprint arXiv:2102.03062* (2021).

[4] Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.

[5] McGuffie, Kris, and Alex Newhouse. "The radicalization risks of GPT-3 and advanced neural language models." *arXiv preprint arXiv:2009.06807* (2020).

[6] Zhang, Min, and Juntao Li. "A commentary of GPT-3 in MIT Technology Review 2021." *Fundamental Research* 1.6 (2021): 831-833.

[7] Wang, Shuohang, et al. "Want to reduce labeling cost? gpt-3 can help." *ArXivpreprintarXiv:2108.13487* (2021).

[8] OpenAI, OpenAI, et al. "Asymmetric self-play for automatic goal discovery in robotic manipulation." *arXiv preprint arXiv:2101.04882* (2021).

[9]Brockman, Greg, et al. "Openai gym." *arXiv preprint arXiv:1606.01540* (2016).

[10] Rae, Jack W., et al. "Scaling language models: Methods, analysis & insights from training gopher." *arXiv preprint arXiv:2112.11446* (2021).

[11] Zhu, Jiongli, et al. "Generating Interpretable Data-Based Explanations for Fairness Debugging using Gopher." *Proceedings of the 2022 International Conference on Management of Data*. 2022.

[12] Zhang, Xingwen, Jeff Clune, and Kenneth O. Stanley. "On the relationship between the OpenAI evolution strategy and stochastic gradient descent." *arXiv preprint arXiv:1712.06564* (2017).

[13] Arroyo, Javier, et al. "An OpenAI-Gym Environment for the Building Optimization Testing (BOPTEST) Framework." *Proceedings of the 17th IBPSA Conference*. 2021.