Text Generator using GPT2 Model

R Roshini , A Preethi , C Arulmozhi and P Sivaganga*

Artificial Intelligence and Data science
Veltech Hightech Dr.Rangarajan Dr.Sakunthala Engineering College,Avadi,
Chennai 600062- India.

Abstract- **GPT-2 is state of art algorithm transfer learning with respect to nlp task we can do it like text summarizationand many more applications. The text generation application programming interface is supported by a large-scale unsupervised language model capable of generating paragraphs of text This transformer-based language model, based on OpenAI's GPT-2 model, accepts a sentence or partial sentence as input and predicts subsequent text. GPT-2 uses input textto set the initial context for further text generation. The length of an input string can range from few words to a maximumsequence length of 1024 tokens. The longer an initial input, the more subject context is provided to a model. Generally, longer inputs produce a more coherent text output. It was specifically trained to guess the next word in sentences. GPT-2 isa massive model that contains a massive amount of compressed knowledge from various parts of the internet. It can be usedto forecast the likelihood of a sentence. The model learns an internal representation of the English language, which it can then use to extract features useful for subsequent tasks. We have used a tkinter which is a GUI to display the generated output.**

Keywords—Text generation, Graphical User Interface, gpt2 transformer,Open AI,tkinter,

1.INTRODUCTION

GPT2 (Generative Pre-trained Transformer 2) Algorithm is an unsupervised transformer language model. Transformer language models cash in of transformer blocks. These blocks make it possible to process intra-sequencedependencies for all tokens during a sequence at an equivalent time. GPT2 may be a powerful generative NLP model that excels in processing long-range dependencies and it's pre-trained on a various corpus of text. Long before the deep learning boom, text generation models were being developed. The thing of these models is to be suitable to prognosticate a word or sequence of words given a textbook. Beforehand textbook generation models were trained using Markov chains, in which each word was a state of the chain, and the probability of the coming word (groundedon the former one) was calculated grounded on the number of circumstances of both words successively within the training textbook. GPT-2 could be a large transformer-based language model with 1.5 billion parameters that has beentrained on a dataset. We scraped content from the web to create a replacement dataset that emphasises diversity of content.

As a result of the size of the dataset, GPT-2 is trained with a simple goal: predict the next word given all of the previous words. Since this goal includes current demonstrations of many tasks across a variety of domains, this goal includes many current tasks from different domains. As a direct scale-up of GPT, GPT-2 features ten times as many parameters and a thousand times as much data as GPT.

II. RELATED WORKS

Antoine Chaffin, Vincent Kijak, Ewa, Claveau, Antoine Chaffin, Proposed Support for large language models(LM)In this study, we explore how this generation is constantly further controlled at decoding time to fit specific conditions(e.g., being nontoxic, portraying certain feelings, using a given erudite kidney,etc.) without fine- tuning theLM. Mills enable for realistic extended tests. After careful consideration, we formulate combination generation as a process of tree disquisition driven by a discriminator that shows how well the associated expenditure complies with the constraint. In addition to being simpler and less precious to trainer, this system allows for a more precise and dynamic utilisation of the limitation. We give a number of new approaches to examine this generation tree, includingthe Monte Carlo Tree Search(MCTS), which offers theoretical assurances on the

Alvin Chan , Yew Soon Ong, Alvin Chan, Bill Pung, Aston Zhung, and Jie Fu Because of its multitudinous operations, neural controllable textbook generation is a critical area gaining attention. Despite the fact that there's a large body of previous add controllable textbook generation, there's no unifying theme. Throughout this work, we give a relief schema for the channel of the generation process by categorising it into five modules. Controlling attributes during the generation process necessitates changes to those modules. We present a summary of colorful ways used to modulate those modules. We also bandy the advantages and disadvantages of those ways. We also pave the way for the development of new infrastructures grounded on the combination of the modules described in this paper.

Ari Holtzman, Jan deals, Li Du, Maxwell Forbes, and Yejin Choi are among the actors. Despite significant progress in neural kingunge modelling, the most introductory decoding strategy for textbook generation from a language model remains unknown. The empirical observation is that, while using liability as a training objective results in high- quality models for a wide range of language understanding tasks, maximization- grounded decoding styles similar as ray similar end up producing mellow, incoherent, or stuck in expectative circles. To address this, we propose Nucleus Sampling, a simple but effective system for rooting much advanced- quality textbook from neural language models than former decoding strategies. Use approach aids test degenerate by slice from the dynamic nexus of commemorative, which contains the vast maturity of the probability mass.

III. MATERIAL AND METHODS

STEPS INVOLVED IN THE GENERATION OF TEXT

GPT-2 could be a transformer-based, autoregressive language model that performs well on a variety of language tasks, particularly (long form) text generation. The model accomplishes this through the use of attention. It enables the model to specialise in words relevant to predicting subsequent words. The Hugging Face Transformers library contains everything you need to coach transformers models.
GPT-2 model:

- Input sentences are given within the prompt
- Load Tokenizer and Data Collator
- Load and setup the Training Arguments
- Generate text with Pipeline

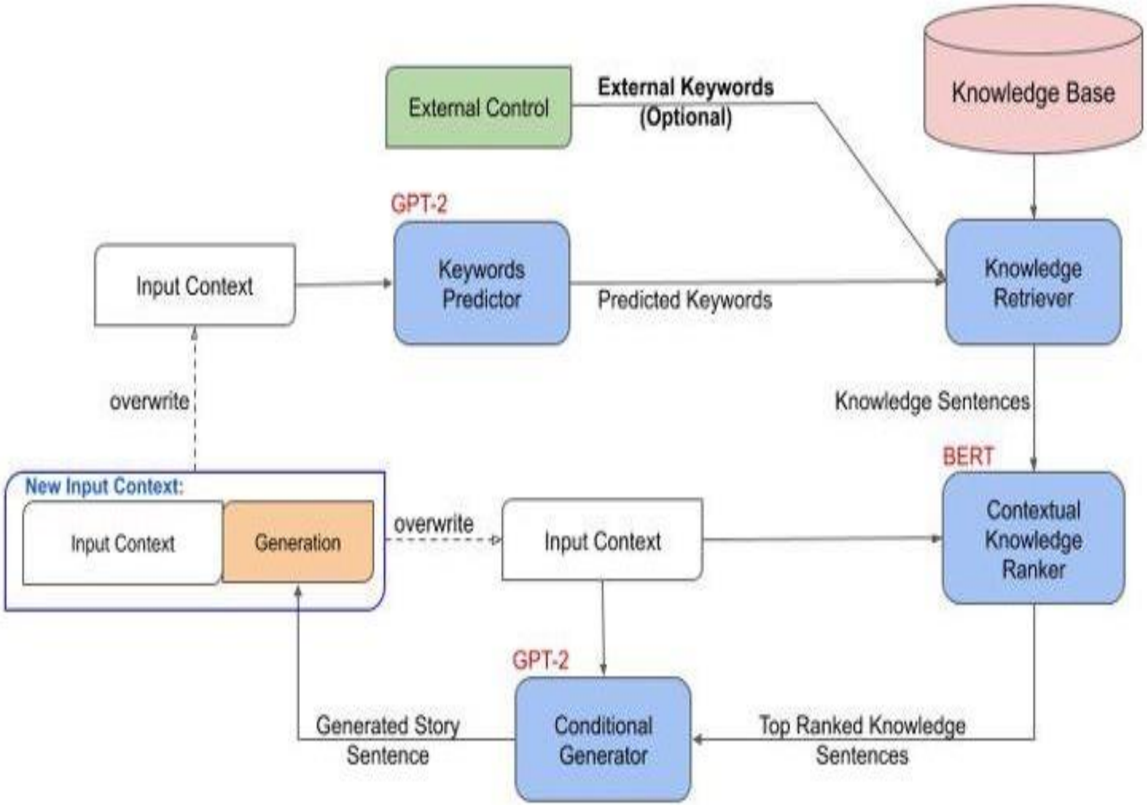
OVERVIEW OF SYSTEM FRAMEWORK

The overview of the research framework for design and development of the proposed automated essay scoring using on NLP with the automated generation of essay. The architecture of the research framework for design and development of the proposed study.
steps:
we need to make sure libraries are installed such as pytorch and transformers

After installing libraries, we are importing its pipeline module The pipeline module is an abstraction layer that takes away the complexity of code and allows an easy way of performing different NLP tasks constructing the GPT-2 text generation pipeline, which uses the most widely used decoder-based transformer paradigmfor language production.

we can start defining the prefix text we want to generate from After we

define our starting text, now it’s time to do generation.



SOFTWARE DEPENDENCIES

PYTORCH

The Torch library and the Python programming language are both supported by the open source machine learning (ML) framework PyTorch. It is among the most widely used venues for comprehensive literacy study. The framework is designed to accelerate systematic research between prototyping and deployment. Similar to NumPy, PyTorch computes with tensors that are sped up by discrete graphics processors (GPUs). Tensors are multidimensional arrays that may be controlled and modified using APIs. Over 200 unique fancy functions are supported by the PyTorch framework. PyTorch continues to grow in popularity because i It makes it easier to create models for artificial neural networks (ANNs). PyTorch is mostly used for functions related to discovery, data mining, and artificial intelligence.

KEY FEATURES OF PYTORCH

Drug users can switch between modes without any interruption thanks to TorchScript, which is widely used as PyTorch's assembly environment.

TorchScript enhances rigidity, speed, usability, and functionality. Dynamic graph calculation: Drug users are permitted at this phase.instead of waiting for the entire Automatic isolation law to be implemented, to change network geste on the cover.

Because PyTorch quantitatively calculates the expansion of a function by doing backward passes in neural networks, this technology supports Python. is built on Python and widely used with well-known packages and libraries like NumPy, SciPy, Numba, and Cynthon

REGEX

A pattern matching mechanism used in programming is called a regular expression. Matching textbook strings is made simple and flexible via regular expressions. Regular expressions are employed to find an algorithmic match to a user's search query in search engines like Google, data validation systems, and syntax checking systems. Regex or regexp are other abbreviations for regular expressions.

The operations that aid in the construction of regular expressions are as follows: Quantification:

Quantifiers specify how frequently the antedating element may occur.Hiatuses can be used to

specify a driver's compass and priority.

Boolean Conditions For drivers and groups, an OR or AND condition is frequently stated.

To match a string, regular expressions use algorithms such as Deterministic Finite Robotization (DFA) and Non- deterministic Finite Robotization (NFA). In an NFA, there are several possible coming countries for each brace of state and input symbol, whereas a DFA accepts a finite string of symbols.

CREATING THE INTERFACE

In the interface we will having one output text box and then generate text will be function. our output title will be TEXTGEN. This is just like a sentence generation and we are going to generate the sentence. We are giving the inputas a word or a sentence in input prompt then and the output will be generated in the gui web application we can able to copy the text and can also generate more paragraph by the generated sentence.

EXPLORE USE CASES AND MODEL PARAMETERS

We can tweak the following model parameters to influence the serving behavior of the model.

--text : sentence in the first place.

--quiet : not print all of the unnecessary information, such as the "==========="

--nsamples : batch size of samples while using a multinomial

--unconditional : Unconditional generation, if true.

--batch_size : various batch sizes

--length : length of the sentence (number of contexts)

--temperature: the distribution of the thermodynamic temperature (default 0.7)

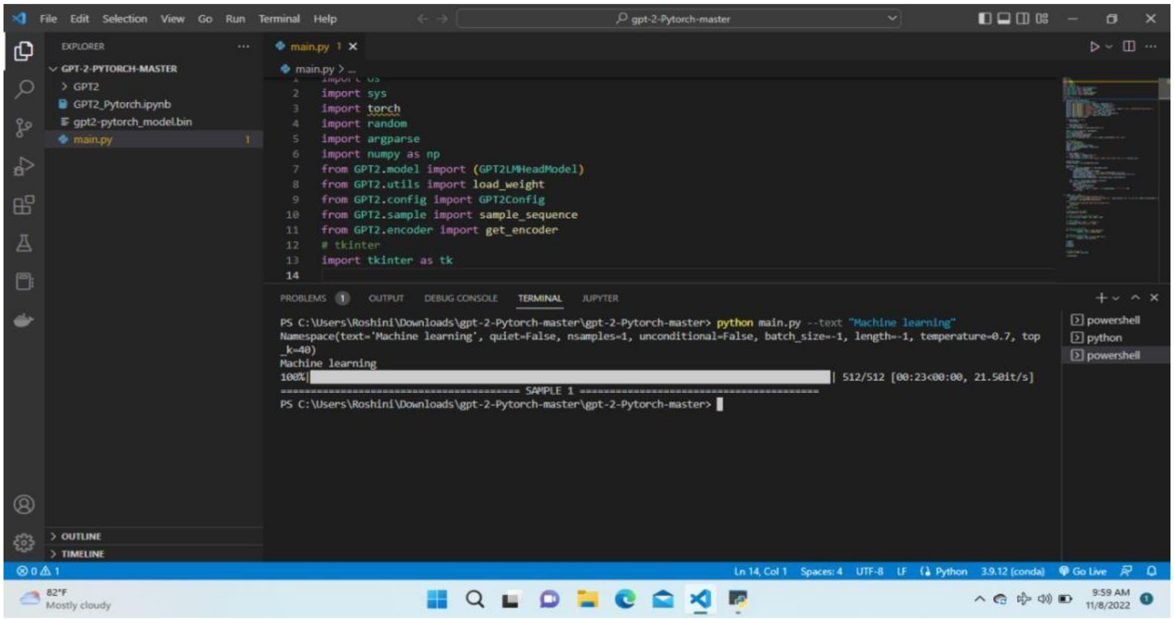
--top_k : Returns the top k biggest elements along a specified dimension of the input tensor. (Standard 40)

IV. RESULT AND DISCUSSION

GETTING INPUT:

We can run a code and get the input with the command $ python main.py --text. Now we have a sample in this we have given the input text as “Machine Learning”.

INPUT TEXT:"Machine Learning"

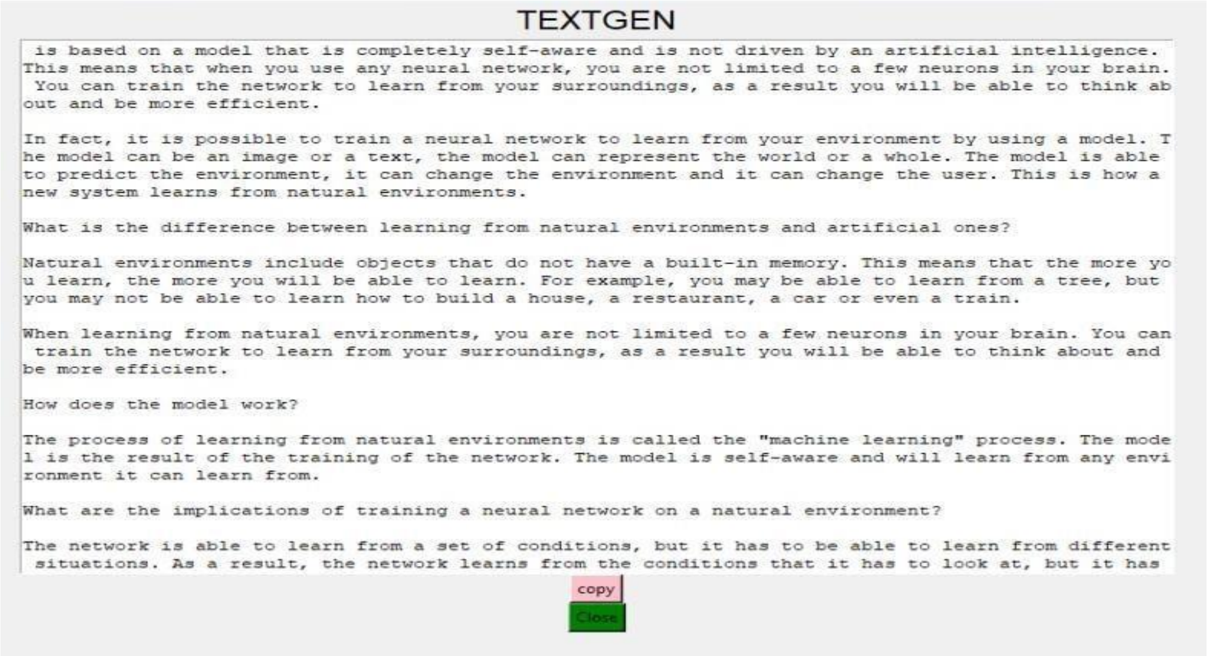


```python
import os
import sys
import torch
import random
import argparse
import numpy as np
from GPT2.model import (GPT2LMHeadModel)
from GPT2.utils import load_weight
from GPT2.config import GPT2Config
from GPT2.sample import sample_sequence
from GPT2.encoder import get_encoder
# tkinter
import tkinter as tk
```

```
PS C:\Users\Roshini\Downloads\gpt-2-Pytorch-master\gpt-2-Pytorch-master> python main.py --text "Machine learning"
Namespace(text='Machine learning', quiet=False, nsamples=1, unconditional=False, batch_size=-1, length=-1, temperature=0.7, top
_k=40)
Machine learning
100%|██████████████████████████████████████████████████| 512/512 [00:23<00:00, 21.50it/s]
======================================== SAMPLE 1 ========================================
PS C:\Users\Roshini\Downloads\gpt-2-Pytorch-master\gpt-2-Pytorch-master>
```

GENERATION OF OUTPUT

Fig 7.6 Output screenshot



```
 is based on a model that is completely self-aware and is not driven by an artificial intelligence.
This means that when you use any neural network, you are not limited to a few neurons in your brain.
 You can train the network to learn from your surroundings, as a result you will be able to think ab
out and be more efficient.

In fact, it is possible to train a neural network to learn from your environment by using a model. T
he model can be an image or a text, the model can represent the world or a whole. The model is able
to predict the environment, it can change the environment and it can change the user. This is how a
new system learns from natural environments.

What is the difference between learning from natural environments and artificial ones?

Natural environments include objects that do not have a built-in memory. This means that the more yo
u learn, the more you will be able to learn. For example, you may be able to learn from a tree, but
you may not be able to learn how to build a house, a restaurant, a car or even a train.

When learning from natural environments, you are not limited to a few neurons in your brain. You can
 train the network to learn from your surroundings, as a result you will be able to think about and
be more efficient.

How does the model work?

The process of learning from natural environments is called the "machine learning" process. The mode
l is the result of the training of the network. The model is self-aware and will learn from any envi
ronment it can learn from.

What are the implications of training a neural network on a natural environment?

The network is able to learn from a set of conditions, but it has to be able to learn from different
 situations. As a result, the network learns from the conditions that it has to look at, but it has
```

V. CONCLUSION

When an outsized language model is trained on an outsized and different dataset, it can perform well across a good range of disciplines and datasets. GPT- 2 achieves slice- edge performance on tested language modelling datasets. The model's capability to perform a colourful set of tasks during a zero- shot selling suggests that high- capacity models trained to maximise the liability of sufficiently varied textbook corpus begin to find out the way to perform a surprisingnumber of tasks without unequivocal supervision.

FUTURE ENHANCEMENT

Methods for detecting machine generated tests typically specialise in binary classification of human versus machine transcription. Misclassification has the potential to harm authors within the scientific domain, where publishers may use these models to look at manuscripts under submission. Additionally, authors may use text generation models appropriately, like with the utilization of assistive technologies like translation tools. during this case, a binary classification scheme might be wont to flag appropriate uses of assistive text generation technology as simply machinegenerated, which might be problematic

REFERENCES

(1) Alvin Chan, Yew- Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2021. Cocon A tone- supervised approach forcontrolled textbook generation. In International Conference on Learning Representations

(2) Ari Holtzman, Jan deals, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural textbookdegeneration. In International Conference on Learning Representations

(3) Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, and NazneenFatemaRajani. 2020. Gedi Generativediscriminator guided sequence generation. CoRR, abs/2009.06367.

(4) Yogatama,D., d'Autume,C.d.M., Connor,J., Kocisky,T., Chrzanowski,M., Kong,L., Lazaridou,A., Ling,W., Yu,L.,Dyer,C., et al. Learning and assessing general verbal intelligence. arXiv preprint arXiv1901.11373, 2019.

(5) Recht,B., Roelofs,R., Schmidt,L., and Shankar,V. Do cifar- 10 classifiers generalize to cifar- 10? arXiv preprintarXiv1806.00451, 2018.

(6) Peters,M.E., Neumann,M., Iyyer,M., Gardner,M., Clark,C., Lee,K., and Zettlemoyer,L. Deep contextualized wordrepresentations. arXiv preprint arXiv1802.05365, 2018

(7) Vinyals,O., Fortunato,M., and Jaitly,N. Pointer networks. In Advances in Neural information wisdom Systems,pp.2692 – 2700, 2015.

(8) Bajgar,O., Kadlec,R., and Kleindienst,J. Embracing data cornucopia Booktest dataset for reading appreciation.arXiv preprint arXiv1610.00956, 2016.

(9) Barz,B. and Denzler,J. can we train on test data? purifying cifar of near- duplicates. arXiv preprint arXiv1902.00423, 2019.