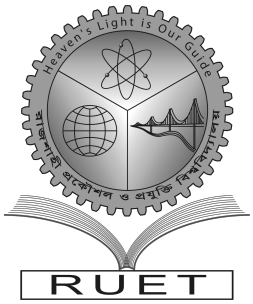
Heaven's Light is Our Guide



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

Generation of Bangla Literature Texts by Fine-tuning Generative Pre-trained Transformer

Author

Md. Abdur Rakib Mollah

Roll No. 1703115

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

Supervised by

Dr. Md. Al Mehedi Hasan

Professor

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

ACKNOWLEDGEMENT

Firstly, I would thank the Almighty for making me successful in creating this thorough thesis book. It would have been impossible to complete the task without His heavenly grace.

I intend to express my gratitude and show my admiration and respect to **Dr. Mir Md. Jahangir Kabir**, Professor, Department of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Rajshahi. He helped me select this thesis topic and introduced me to the key ideas underlying my research. I deeply appreciate the guidance and mentorship I received from him. His insightful advice and encouragement not only helped me select an engaging thesis topic but also provided me with a solid understanding of the foundational concepts that underpin my research.

But since he had to go abroad, this thesis work was supervised by **Dr. Md. Al Mehedi Hasan**, Professor, Department of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Rajshahi. He has not just given me the technical directions, advice, and documents required to carry out the thesis over the course of the year, but he has also consistently motivated me, provided me with advice, extended help, and collaborated supportively whenever he deemed appropriate. His unwavering aid was the most efficient tool I had to accomplish my tasks. He was ever-present for me whenever I encountered any complex challenges or situations, irrespective of the hour. Without his sincere care, this thesis would not have assumed its final shape.

Additionally, I wish to extend my most heartfelt thanks to the teachers of Computer Science & Engineering, Rajshahi University of Engineering and Technology for dedicating their time, expertise, and hard work to foster an environment conducive to academic research.

In conclusion, my profound appreciation goes out to my exceptional parents for their unwavering support and encouragement throughout this journey.

August 14, 2023
RUET, Rajshahi

Md. Abdur Rakib Mollah

Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

CERTIFICATE

This is to certify that this thesis report entitled ***"Generation of Bangla Literature Texts by Fine-tuning Generative Pre-trained Transformer"*** *submitted by* ***Md. Abdur Rakib Mollah, Roll:1703115*** *in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Department of Computer Science & Engineering of Rajshahi University of Engineering & Technology, Bangladesh is a record of the candidate own work carried out by him under my supervision. This thesis has not been submitted for the award of any other degree.*

Supervisor

_____

Dr. Md. Al Mehedi Hasan
Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

External Examiner

_____

Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

ABSTRACT

The field of Natural Language Generation (NLG) has seen a significant transformation in text generation across many fields with the emergence of sophisticated models such as GPT-2. The objective of this research is to assess the efficacy of GPT-2 in producing text that is both logically connected and contextually appropriate. The dataset used for the research is completely coherent and collected from authentic sources by web scraping. The evaluation of GPT-2's outputs will be conducted using a dual approach, including human assessment and the use of the Flesch Reading Ease Score. Utilizing the potential of GPT-2, we initiated an extensive investigation into its aptitude for generating language. To implement the model within hardware limitations, some modifications were made. A wide array of text samples was developed via rigorous testing, covering several disciplines such as artistic writing and technical documentation. In order to assess the quality of these results, a two-fold assessment technique was used. The technique used in this study was founded upon the fundamental principle of human assessment. The created text was carefully evaluated by skilled human assessors, who thoroughly examined its fluency, coherence, and relevancy based on predetermined criteria. In addition to the human review, we used the Flesch Reading Ease Score as a quantitative measure of the readability of the produced text. Through the use of a scoring system that takes into account factors such as sentence length, syllables per word, and characters per word, we have obtained an impartial metric that quantifies the level of complexity and ease of comprehension shown by the given text. This enabled us to determine if the produced information is consistent with the intended standards of readability. The results obtained from our research provide a comprehensive understanding of the many dimensions of GPT-2's natural language generation capabilities. Most of the generated texts were easier to read and by human evaluation, the texts were indistinguishable from human-written texts on 80.76% of cases. In summary, our research highlights the need of integrating human assessment with quantitative measurements, such as the Flesch Reading Ease Score, to holistically evaluate the effectiveness of GPT-2 in natural language generation tasks. This methodology offers a comprehensive comprehension of the model's capabilities and constraints, hence helping the continuous progress of AI-driven language synthesis in many domains.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Introduction

Natural language generation (NLG) is well recognized as a prominent method used in Natural Language Processing (NLP) [1]. Advanced language models like GPT (Generative Pre-trained Transformer) is being used for generative tasks in the field of English literature. Similarly, modified versions of this GPT model is used for poetry or similar literature generative tasks in other languages also like Arabic, Chinese, French. Sophisticated linguistic models and methodologies employed for the purpose of generating content in the realm of literature and creative writing. Long Short-Term Memory (LSTM) networks are a recurrent neural network (RNN) architecture that has gained significant popularity and extensive use across several domains of research and practical implementations.[2]LSTM networks are a specific kind of recurrent neural networks (RNNs) that have been designed in particular to efficiently collect and represent extensive relationships present in textual input. Machine learning models have been employed for the purpose of generating poetry, completing text, and engaging in creative writing endeavors. Researchers have conducted investigations into the application of transformative models, such as GPT, in the realm of poetry generation. The aforementioned models has the capability to acquire knowledge of many literary styles, rhyme schemes, and meter in order to produce original poems.

There have been several endeavors that have especially concentrated on employing machine learning methods to generate sonnets in the form of Shakespearean sonnets.[3] The models have undergone training using Shakespearean sonnets in order to replicate his distinctive style and linguistic patterns. Character-level text generation involves the generation of text at the

level of individual characters, as opposed to generating text based on words or tokens. This has the potential to yield intriguing and innovative outcomes, such as the generation of textual content that emulates the distinctive style of a particular character. Markov chains are stochastic models that possess the ability to generate textual content by making predictions for the subsequent word, relying on the present word as a basis[4]. Although less complex compared to contemporary brain models, they are nonetheless capable of generating innovative and unforeseeable outcomes. Hybrid methodologies are employed in certain projects wherein a combination of diverse techniques is utilized, exemplified by the integration of manually designed rules in conjunction with machine learning models. This amalgamation is undertaken with the aim of attaining particular objectives in the realm of creative writing. The procedure of generating text relevant to a certain genre includes the training of models using diverse genres, such as science fiction, fantasy, or mystery. This methodology allows the models to generate content that conforms to the unique characteristics and norms of each genre. The advent of online platforms has facilitated the practice of collaborative storytelling when individuals engage in a sequential process of adding to a shared narrative. Algorithms have the potential to facilitate the integration of story elements in order to create a cohesive narrative.

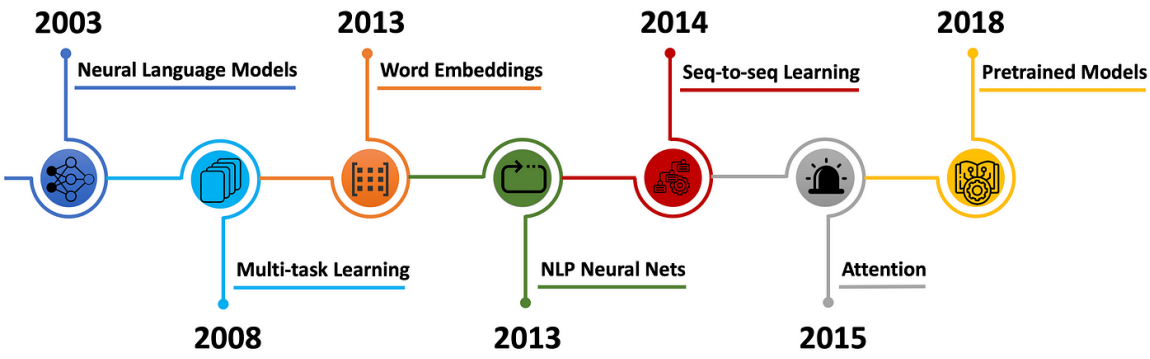


Figure 1.1: Evolution of Pretrained Models [Image taken from internet]

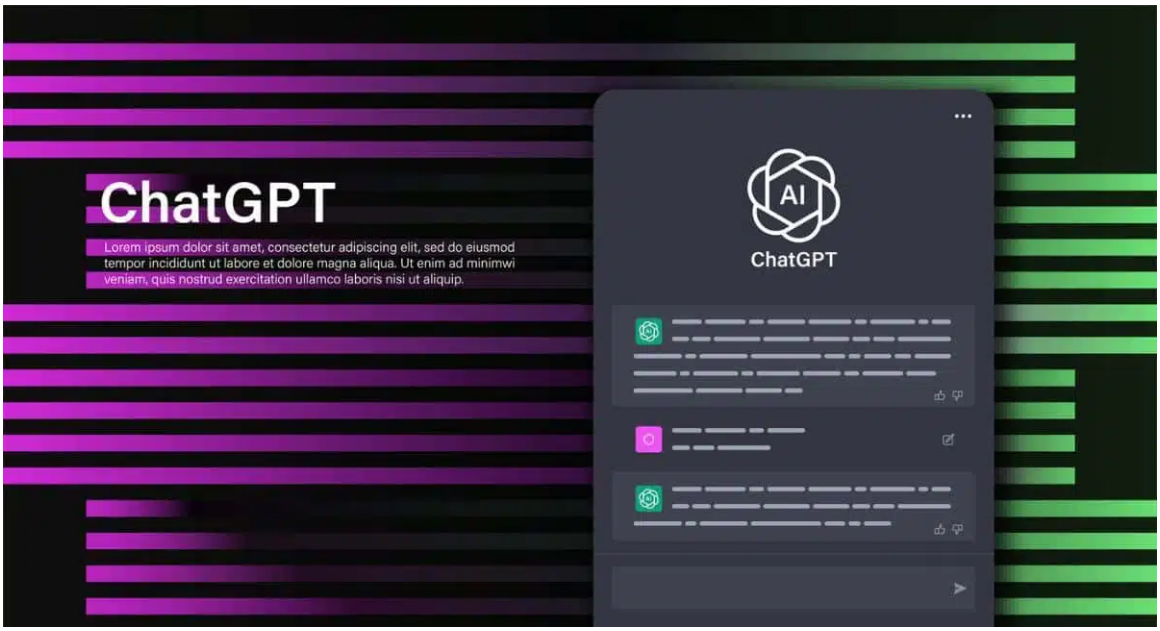


Figure 1.2: Application Of NLG (Collected from internet).

1.2 Problem Identification

The The field of Natural Language Generation (NLG) in the Bengali language, also known as Bangla, has unique challenges stemming from its complex linguistic features and the scarcity of available resources.[5] The obstacles involve a wide range of elements, including language-specific characteristics and technological limits. One of the primary obstacles that emerges in the creation of dependable natural language generation (NLG) systems is the restricted accessibility to comprehensive resources, such as annotated corpora and pre-trained models, which have substantial importance. In contrast to the abundance of well-established resources available for English, the Bangla language lacks comparable support, which poses challenges to the creation and execution of natural language generation (NLG) technologies. The problem of morphological complexity poses an additional concern. The complex system of inflections and derivations in the Bangla language needs meticulous treatment in order to maintain grammatical precision and produce output that sounds genuine. The flexibility of word order in the language also presents obstacles, since the accurate arrangement of words plays a significant role in effectively transmitting exact meanings[6]. Furthermore, the challenge of reliably identifying named entities persists owing to the limited availability of comprehensive named entity

databases. This limitation has a significant influence on jobs that require the generation of contextually appropriate material. Capturing the complex expressions of mood and tone in Bangla writing is a significant challenge, but it is crucial for the development of successful content. There are ongoing endeavours being made to address and alleviate these difficulties. The development of customised linguistic resources and models is being propelled by collaborative efforts among linguists, academics, and technologists. Notwithstanding the challenges, the resolution of these difficulties has the potential to boost natural language generation (NLG) skills in the Bangla language and facilitate the development of more advanced language generation systems.

The primary objective of this thesis research is to design a methodology that facilitates the effective and skillful production of high-quality textual data using Natural Language Generation (NLG) models. The dataset used for the purposes of training and creation will consist of literary content. The main aim is to faithfully replicate the essential attributes and wide-ranging properties of real-world data, including intricate patterns, distributions, correlations, and concerns pertaining to confidentiality.

Furthermore, a complete assessment system has been suggested to guarantee the reliability and usefulness of the artificially created data. The purpose of this evaluation method is to verify the genuineness and practical use of the synthetic data, highlighting its dependability and relevance for natural language generation (NLG) activities in diverse settings.

1.3 Motivation

The basic and fundamental purpose of this thesis was to collect a consistent dataset including textual material from the literary sphere. This phase laid the groundwork for the succeeding stages of study.

The second reason included carefully selecting and implementing an appropriate pre-trained language model that had been particularly trained on Bangla texts. This choice was motivated

by the need to match pre-training data with the kind of data that would be fine-tuned to obtain best outcomes.

The third reason was to apply and investigate various assessment methodologies targeted at gauging the quality, coherence, and effectiveness of the created texts. These strategies were used to guarantee a thorough and multifaceted evaluation of the created information.

This thesis seeks to improve the understanding and capabilities of generating high-quality Bangla textual content in the context of literature by incorporating careful dataset curation, tailored language model deployment, and comprehensive evaluation strategies.

1.4 Goals

The study's overarching goal is to enhance the production of literary-quality prose in Bangla. A unified corpus of literary works is the desired end result of this endeavour. The next step is to apply a pre-trained language model to Bangla texts that is most appropriate for the task at hand. Finally, to put into practise and investigate a range of assessment strategies for gauging the accuracy, consistency, and usefulness of the created texts. The successful completion of this thesis will increase knowledge and skills necessary to produce high-quality literary writing in Bangla.

1.5 Benefits, Ethics and Sustainability

AI that can come up with new thoughts and answers is called "generative AI." Its ease can speed up the process of making and personalising material, making the user experience better. It also helps improve data in areas where there aren't many records.
To make sure everything is fair, the technology needs to be managed carefully so that it doesn't reinforce any biases that are already in the data. There are risks of misinformation because generative AI could be used to make fake material. It is very important to find a balance between making material and protecting copyright and intellectual property rights.

Complex generative models require a lot of energy to learn, so methods that use less energy are needed. It is very important to divide up resources carefully between AI apps and think about the technology's long-term effects on society and the economy.
It's important for technologists, ethicists, lawmakers, and society as a whole to work together to handle these issues and figure out how to safely integrate generative AI into different fields.

1.6 Research Objectives

The objectives of this thesis are as follows:

(a) The initial step entails gathering a coherent text dataset focused on Bangla literature.

(b) Select a suitable language model that is well-pretrained on Bangla texts.

(c) Fine-tune the selected language model using the collected Bangla literature dataset

1.7 Delimitations

This thesis does not focus on establishing an alternative to larger applications like ChatGPT; rather, it focuses on constructing a process to produce texts in Bangla in a coherent manner, as well as building an adequate evaluation mechanism to evaluate the quality of generated texts. In addition, this thesis also focuses on developing an effective assessment mechanism to measure the quality of generated texts. The focus of the thesis is on constructing a whole paragraph using just one word or one sentence at a time. However, the writing of an entire novel or story is outside the scope of what will be covered in this study.

1.8 Thesis Organization

Chapter 2 - Background Study

A full summary of the background information and historical setting that are important to the

study subject was provided in the chapter.

Chapter 3 - Literature Review

Here, a comprehensive examination of the available literature pertaining to the research subject is undertaken. This task entails the synthesis and critical evaluation of pertinent scholarly articles, research studies, and academic literature that have been already published.

Chapter 4 - Large Language Models

The next chapter is expected to provide an elucidation of the notion of big language models, which include sophisticated artificial intelligence models such as GPT-3. These models have undergone extensive training on vast quantities of textual data in order to produce language that closely resembles human expression.

Chapter 5 - Methodology and Implementation

This chapter describes the study's research methodologies. It might include the methodology, data collecting techniques, instruments, and technologies used experimental design, and other appropriate study processes.

Chapter 6 - Result and Performance Analysis

This chapter presents study findings. Data analysis, graphs, charts, tables, and other visual representations of study findings are examples. The chapter may also analyse findings and their consequences.

Chapter 7 - Conclusion and Future Works

The last section of this study presents a comprehensive overview of the whole research effort.

1.9 Conclusion

This chapter talks about what NLG is, how it can be used, and what kinds of problems it can help solve. We also talked about the problem we were trying to solve, why we were trying to solve it, and what we wanted to accomplish with this thesis work. Along with the limitations, we've also talked about the work's relevancy, ethics, and ability to last. This part gives a general outline of the book and explains why we did our study.

Chapter 2

Background Study

2.1 Introduction

It's helpful to have some familiarity with the areas and evolution of Artificial Intelligence in order to comprehend the work because the contributions touch on a variety of different machine learning subjects. We will now concentrate on some core machine learning and Generative Pre-trained Transformer (GPT) concepts which helps to understand the approach better.

2.2 Artificial Intelligence

Within the broad field of computer science is a subfield known as artificial intelligence (AI), which focuses on the development of intelligent agents that can freely learn, reason, and take action. Research in artificial intelligence has been quite successful in developing effective strategies for coping with a wide range of challenges, ranging from video game play to medical diagnosis. [7].

One of the most challenging obstacles that artificial intelligence research must overcome is the development of intelligent robots with the capacity to gain knowledge via experience. People are only able to improve their performance over time if they have the ability to gain knowledge from their past errors. The development of programs that are capable of gaining knowledge via experience is the focus of the subfield of AI known as machine learning.

Machine learning, or ML for short, is a subset of artificial intelligence that gives computers the ability to learn from data and evolve over time without being specifically programmed to

do so. ML algorithms employ data-driven tactics to uncover patterns and correlations in data, after which they use that knowledge to produce predictions or judgements. These strategies are utilized to find the patterns and correlations.

2.3 Machine Learning

Machine learning, as a constituent of artificial intelligence (AI), facilitates the acquisition of knowledge by computers via the analysis of data, allowing them to generate predictions or make judgments without the need for explicit programming. Artificial intelligence serves as the comprehensive designation for this domain of academic inquiry. The advancement of computer learning necessitates the creation of algorithms and models that facilitate the acquisition of knowledge from past experiences, hence enhancing the computational capabilities of machines. Machine learning algorithms use data-driven methodologies to identify patterns, correlations, and underlying structures within the data, rather than being tailored to specific objectives. Machine learning enables the discovery of insights that were previously concealed.

The training data and the model are the two main elements of machine learning[8]. The algorithm learns from instances that have been labeled during the training phase and modifies its internal parameters to reduce prediction errors. Consequently, the model acquires the capacity to extrapolate from the training dataset and afterward uses this capacity to make predictions for new, unaltered data.

Machine learning may be broadly classified into three main categories: reinforcement learning, unsupervised learning, and supervised learning. Training on labeled data with specified input-output pairings is known as supervised learning. When working with data that has not been labeled, unsupervised learning focuses on locating structures and patterns within the data. [9]. Reinforcement learning allows an agent to interact with its surrounding environment and gain knowledge from the feedback it receives, which might take the form of incentives or penalties. Machine learning has found applications in an extensive range of fields, including natural language processing, computer vision, recommendation systems, medical care, and financial services, amongst others. Its success may be due to its capacity for managing complicated and large-scale data, automating decision-making procedures, and constantly enhancing performance through iterative learning. Machine learning is continuing to change businesses and

propel innovations in many different fields as technology develops.

2.3.1 Supervised Learning

Compared to unsupervised learning, supervised learning exhibits higher control and less bias. In supervised learning, the user gives the algorithmic program pairs of matching input and output. As a result, the program creates a technique to produce the required output depending on input. This algorithmic capacity also includes automatically producing outcomes for unknown inputs. Imagine, for the sake of illustration, that a sizable batch of emails are sent to the algorithmic program along with accompanying signals indicating whether or not each email is categorized as spam (the intended result). The algorithmic program can correctly assess whether a brand-new email qualifies as spam or not, demonstrating the effectiveness of supervised learning.

The usual algorithmic program for supervised machine learning consists of about three components.

(a) **Decision function**: A series of calculations or processes known as a ”decision function” make use of the given data to ”infer” the logical pattern that your algorithmic technique is attempting to find. [10].

(b) **Error function**: If there are proven examples, comparing the estimate to them may be used to determine how accurate it is. Did the method of choice successfully discover the truth? If not, how can the level of accuracy or the scope of the mistake be measured?

(c) **Updating decision**: A technique in which the algorithm’s detection of mistakes affects how decisions are made and their effects on the output, reducing the number of errors in following rounds. [10].

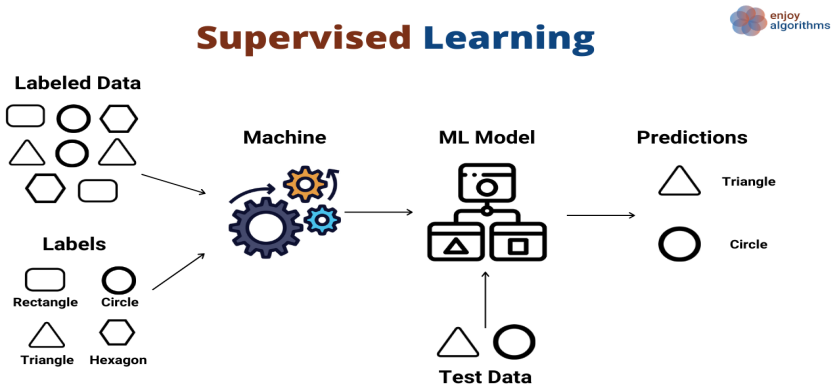


Figure 2.1: Supervised Learning Mechanism(Image taken from internet)

2.3.2 Unsupervised Learning

A key idea in machine learning is unsupervised learning, which moves the emphasis from teaching computers on labeled data to finding hidden structures and patterns in unlabeled datasets. Unsupervised learning aims to glean useful information from data without predetermined labels or goal values, in contrast to supervised learning, where models are directed by labeled instances to anticipate certain outcomes.

During unsupervised learning, the computer program is able to discover inherent connections, groups, and patterns within the data. Clustering, where the algorithm clusters similar data points together based on common properties, is one of the main strategies in this area. This may reveal natural divides in the data, such as unique species in scientific investigations or client groups in marketing.

Figure 2.2: Unsupervised Learning Mechanism (Image taken from internet).

2.3.3 Semi-supervised Learning

Semi-supervised learning is a kind of machine learning that occupies an intermediate position between supervised learning and unsupervised learning [11]. In this method, both labeled and unnamed data are used to train the program, taking advantage of the best of both worlds. Labeled data leads the model's learning process, but adding unnamed data lets the computer find hidden patterns and structures in the data, which improves its ability to generalize.

By using the relationships that already exist in both labeled and unlabeled data, semi-supervised learning tries to get around the problems with supervised learning, which relies a lot on labeled examples, and the exploratory potential of unsupervised learning, which doesn't have any labels to guide it. This method is especially useful when it's hard or expensive to get a lot of data with labels. It's a faster way to train models that can generalize well and make accurate predictions across a wide range of data points.

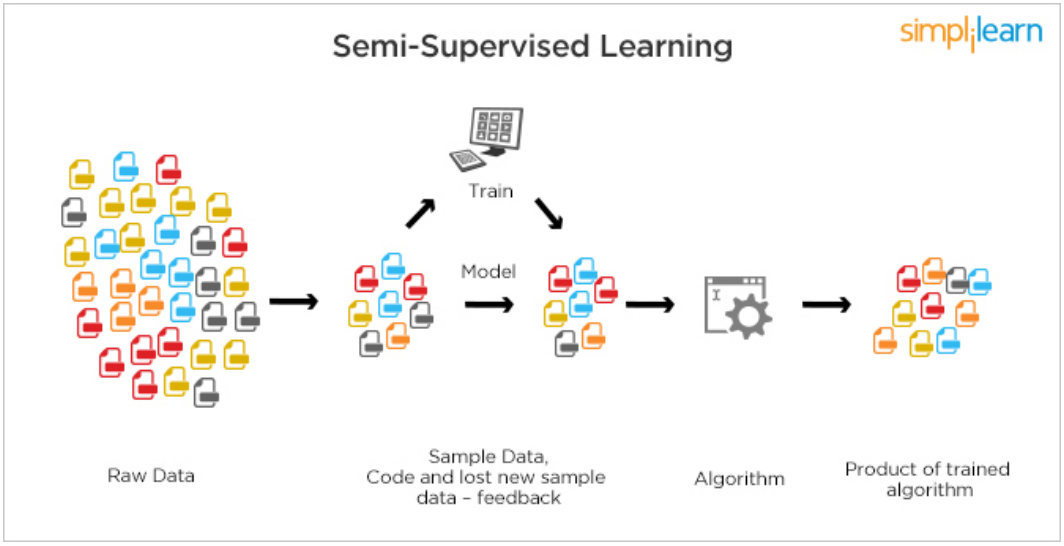


Figure 2.3: Semi-supervised Learning Mechanism (Image taken from internet)

2.3.4 Reinforcement Learning

Reinforcement learning is a machine learning paradigm whereby an autonomous agent acquires the ability to make sequential decisions via iterative interactions with its environment. Through a process of trial and error, the individual tries to maximize a total payout by taking actions that

lead to good results. The agent gets input in the form of awards or punishments based on what it does. This helps it learn the best ways to act. Reinforcement learning involves trying out different things and using what you've learned to find a balance between short-term goals and long-term benefits. This method of dynamic learning is used a lot in robots, games, suggestion systems, and self-driving cars.

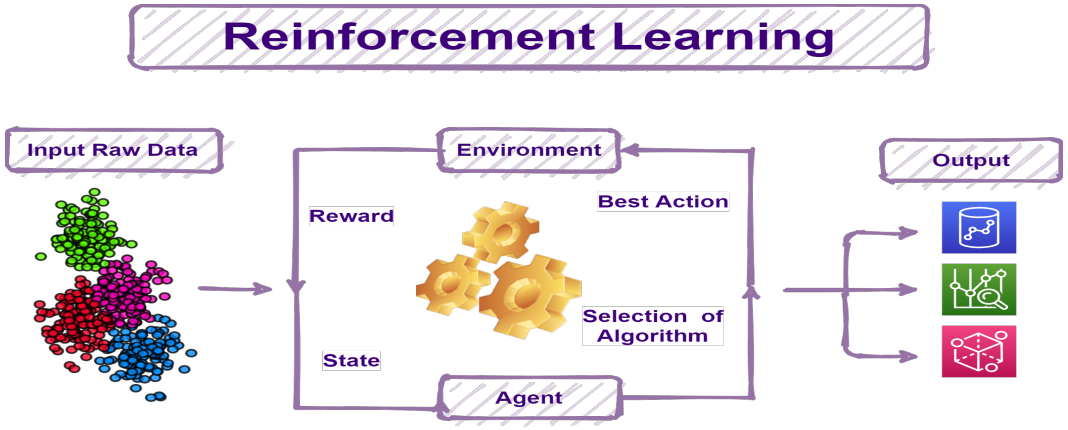


Figure 2.4: Reinforcement Learning (Image taken from internet)

2.4 Artificial Neural Networks

TThe Artificial Neural Network (ANN) has two crucial stages: the forward pass and the backward pass. During the process of the forward pass, input data is sent to the neural network. The network then moves through its hidden layers, using weights and activation functions to make an estimate of what the result will be[12]. The next step is the backward pass, also called backpropagation. It figures out the difference between the mistake of the network and its weights. This gradient tells optimization tools how to change the weights, which improves the accuracy of the network over time. The forward and backward passes work together to help the network learn from data and improve its guesses until they are close to what is wanted.

In the next sub-section, we will discuss about backpropagation and gradient descent.

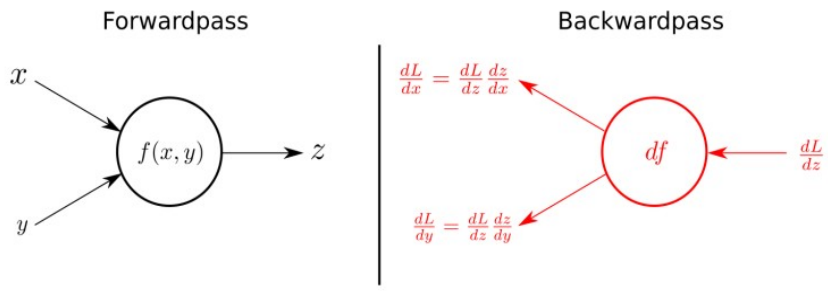


Figure 2.5: Forward Pass and Backward [13]

2.4.1 Backpropagation and Gradient Descent

Two of the most fundamental concepts in machine learning and neural networks are known as backpropagation and gradient descent, respectively. [14]. Backpropagation includes sending mistake information backward through the network's layers to figure out the gradients. It figures out how each network setting affects the general error and lets you make changes to reduce the error. On the other hand, Gradient Descent uses these slopes to change the model's parameters over and over again. It takes charge of the optimization process and modifies the elements in a way that is counter to the gradient in order to locate the least value of the loss function. Backpropagation and Gradient Descent are two ways that neural networks can learn and get better over time.

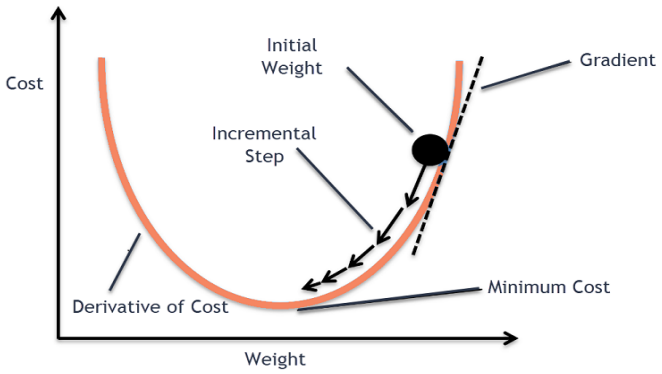


Figure 2.6: Gradient Descent Algorithm [15]

2.5 Recurrent Neural Network

An artificial neural network known as a Recurrent Neural Network, or RNN, is a sort of neural network that is meant to process sequential data by preserving a hidden state that detects and accounts for temporal relationships in the input sequence. Due to the fact that RNNs have feedback connections, as opposed to traditional feedforward neural networks, they are more suited for tasks that need sequences. Some examples of such tasks include processing spoken language, voice recognition, and the study of time series. Because of these links, information is able to withstand the passage of time.

One of the most essential qualities of RNNs is their ability to accommodate inputs of varying lengths over longer periods of time. At each time step, an RNN performs processing on a sequence element, updating its hidden state in accordance with the most recent input as well as the hidden state from the previous time step. Because of the recurring nature of its architecture, RNNs are able to maintain context and keep track of earlier inputs even while they conduct analysis on subsequent parts of the sequence.

Nevertheless, conventional recurrent neural networks (RNNs) have challenges in acquiring long-term dependencies due to the vanishing gradient problem. This issue arises when gradients, used for learning, exponentially diminish as they travel backwards in time. [16]. To address this issue, researchers have developed more advanced versions of Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). These variations have been designed to mitigate the aforementioned difficulty.

, In order to prevent disappearing gradients, LSTM and GRU networks, include gating methods to regulate information and gradient flow[17]. These variations are now the preferred options for assignments requiring sequence modeling since they have shown improved performance in capturing long-term dependencies.

RNNs have made a substantial impact on a diverse array of uses, including sentiment analysis, speech synthesis, music production, and machine translation. Due to computational and memory limits, they do, however, also have difficulties when processing extremely lengthy sequences. Because of this, more modern architectures, such as Transformers, have come to be used as substitutes for some sequence-to-sequence tasks while still maintaining the important contributions made by RNNs to the area of deep learning.

2.6 Deep Learning

Deep learning involves a series of interconnected layers that are capable of autonomously learning complicated mappings quickly and effectively. Through its hidden layer design, a deep learning technique progressively learns classes, initially developing low-level classes like as letters, then relatively higher-level classes such as words, and lastly higher-level classes such as sentences [18].

Deep neural networks consist of several layers of nodes that are interconnected, whereby each layer leverages the information from the preceding layer to enhance the accuracy of prediction or classification [18]. The phenomenon of information flow inside the network, following a sequential order of processing, is often referred to as forward propagation. The observable strata of a deep neural network include the input and output layers. The deep learning model gets the necessary data in the input layer, and generates the final prediction or classification in the output layer[18].

2.7 Deep Generative Models

Unsupervised learning is used to create a generative model, which may be used to learn any form of data distribution [19]. Generative models aim to acquire knowledge of the underlying data distribution of the training set in order to generate novel data points with varying characteristics. Deep generative models refer to multilayer neural networks that have the ability to generate samples that align with the underlying distribution of the data [19].

Deep generative models are neural networks with numerous hidden layers that have been trained using samples to approximate complex, high-dimensional probability distributions. Once the model has been sufficiently trained, it may be used to estimate the probability of each observation and to create new samples from the original distribution [20].

GPT may be classified as a generative model due to its capacity to generate text by using the knowledge acquired during its pre-training phase, whereby it assimilates information from an extensive corpus of textual data. The model's ability to generate language is enhanced by its capability to produce text that is both cohesive and fluent. This is achieved by the model's

training in predicting the subsequent word in a sequence, taking into account the contextual information provided by the preceding word.

2.8 Transformer

The Transformer model is a ground-breaking example of deep learning architecture that has made important contributions to the natural language processing (NLP) industry. The concept was first presented in the scholarly article published in 2017- ”Attention Is All You Need” by Vaswani et al., the Transformer model has become the backbone of numerous state-of-the-art NLP applications due to its ability to effectively handle long-range dependencies in sequential data.

In contrast to conventional sequential models such as Recurrent Neural Networks (RNNs), the Transformer model utilizes self-attention mechanisms to effectively capture the interdependencies across various points within the input sequence. This strategy facilitates the model in evaluating the significance of individual input elements during prediction, hence allowing it to effectively handle lengthy sequences without encountering the issue of vanishing gradients that was prevalent in prior systems based on recurrent neural networks (RNNs).

The architecture of the Transformer has two primary elements: an encoder and a decoder. The encoder and decoder are composed of numerous layers, each including two key sub-modules: the multi-head self-attention layer and the feedforward neural network layer.

In the self-attention layer, the model calculates the attention scores between each input element and every other element in the sequence, determining their relevance. The attention scores are then used to compute weighted sums of the input sequence, emphasizing critical information and suppressing irrelevant parts. This mechanism allows the Transformer to capture both local and global dependencies in the input sequence effectively.

The feedforward neural network layer further processes the outputs of the self-attention layer through point-wise fully connected layers and non-linear activation functions, adding depth and complexity to the model’s representations.

The Transformer model’s self-attention and feedforward layers can be stacked multiple times, leading to the concept of ”deep” Transformers, capable of learning complex patterns and representations from vast amounts of data. The profound achievements of deep Transformers have

been shown in a diverse array of natural language processing (NLP) endeavors, including machine translation, sentiment analysis, question-answering, language production, and several others.

One of the fundamental attributes of the Transformer model is in its capacity to undergo pre-training on large collections of textual data via unsupervised learning, therefore acquiring comprehensive language representations. The pre-existing model may then undergo fine-tuning on targeted downstream tasks using comparatively little labeled datasets, so rendering it exceptionally efficient and efficacious for diverse natural language processing (NLP) applications.

summary, the self-attention mechanism of the Transformer model, in conjunction with its capacity to capture extensive dependencies, has brought about a paradigm shift in the field of natural language processing, leading to substantial enhancements in the efficacy and effectiveness of diverse language-oriented undertakings. The ongoing influence of this phenomenon persists in molding the development of sophisticated natural language processing (NLP) models and propelling advancements in artificial intelligence (AI) research.

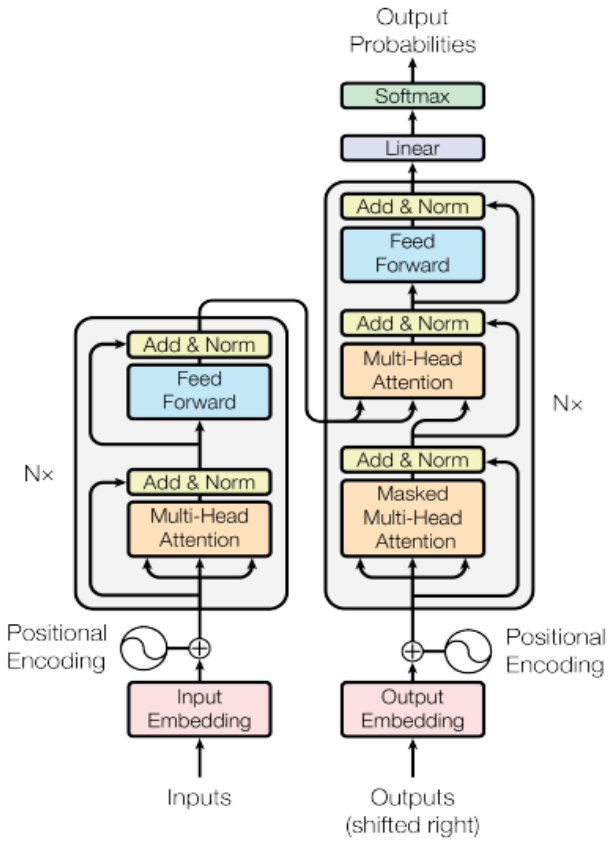


Figure 2.7: Transformer Architecture

2.9 Pre-trained Models

Pre-trained models are a class of machine learning models that have been pre-trained on vast amounts of data to learn rich and meaningful representations of the data[21]. These models have been trained on large-scale datasets using powerful hardware and computational resources, enabling them to capture complex patterns and structures present in the data. Pre-training typically involves unsupervised learning, where the model learns to predict or reconstruct the input data without relying on explicit labels or annotations.

The pre-training process is a crucial step in the development of deep learning models, especially in natural language processing and computer vision tasks. By pre-training on large and diverse datasets, the model can learn general features and representations that are transferable across various specific tasks.

In the context of natural language processing, pre-trained models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) have revolutionized the field. These models are pre-trained on massive text corpora and can understand the context and meaning of words and sentences, making them highly useful for tasks such as sentiment analysis, named entity recognition, question-answering, and text generation.

In computer vision, pre-trained models like VGG (Visual Geometry Group)[22], ResNet[23] (Residual Neural Network), and EfficientNet[24] have become popular choices. These models are pre-trained on large image datasets, allowing them to recognize objects, shapes, and visual patterns effectively. Fine-tuning these pre-trained models on specific tasks like image classification or object detection significantly reduces the amount of data and time required for training.

The use of pre-trained models offers several advantages. First, it saves time and computational resources since pre-training is done once on a large dataset, and the model can be reused for various downstream tasks. Second, pre-trained models can achieve better performance than training from scratch, especially when the task has limited data available. Third, pre-trained models can act as knowledge transfer mechanisms, as the learned representations can be fine-tuned for specific tasks, even in domains with less available data.

However, pre-trained models also have some limitations. They may not be optimized for specific tasks or domains, and fine-tuning requires additional labeled data for the specific task, which might not always be available. Additionally, pre-trained models can be computationally expensive to deploy in production systems, especially if they have a large number of parameters.

In conclusion, pre-trained models have become indispensable tools in modern machine learning, offering significant benefits in terms of performance, transferability, and reduced training time. As researchers continue to develop more sophisticated architectures and datasets, pre-trained models will likely play an even more significant role in advancing various AI applications across different domains.

2.10 Generative Pre-trained Transformer (GPT)

The Generative Pre-trained Transformer (GPT) mechanism is a significant breakthrough in the fields of natural language processing and artificial intelligence. The GPT model, which was created as a result of OpenAI's pioneering research[25], presents a revolutionary methodology for generating language. This is achieved via the use of deep learning techniques and transformer structures.

The fundamental basis of the GPT mechanism is a framework that involves pre-training and fine-tuning[26]. In the first training step, a substantial volume of textual data is used to build a multi-layered transformer model. The aforementioned model is designed to acquire the ability to anticipate the subsequent word in a given phrase, therefore including complex patterns, grammatical principles, and contextual associations inherent in the language. The attention mechanism of the transformer architecture is of particular importance as it facilitates the comprehension of contextual intricacies between words within a sentence, hence empowering the model to produce content that is cohesive and contextually suitable.

After the first pre-training phase, the GPT model proceeds to undergo fine-tuning on task-specific objectives. This entails the process of training the model on more specific datasets and then modifying it to effectively execute tasks such as language translation, text completion, and even creative writing. The GPT's adaptability is shown by its capacity to produce text that closely resembles human language, making it a desirable instrument for many purposes like as content creation, virtual assistants, and other applications.

The success of the GPT mechanism may be attributed to its extensive size, which enables it to acquire complex language patterns and delicate semantic nuances[27]. Through the process of training on a multitude of varied sources of data, GPT is able to include a broad range of situations and phrases, establishing itself as a flexible and adaptable language generator. The

progress made in this field has allowed the development of successive versions, each characterized by larger size and enhanced capabilities, exemplified by GPT-2 and GPT-3.

2.11 Natural Language Generation

Natural Language Generation (NLG) is a branch of natural language processing (NLP) that focuses on automatically producing human-like text from structured data or other forms of input. NLG systems utilize various techniques, including rule-based approaches, template-based methods, and more advanced deep learning models.

In rule-based NLG, linguistic rules and templates are used to map input data to corresponding text segments. This approach is straightforward but limited in handling complex and dynamic language generation tasks[28].

Template-based NLG employs pre-defined templates with placeholders that are filled with relevant data. While more flexible than rule-based methods, template-based NLG still has constraints due to fixed patterns[29].

Recent advancements in deep learning have significantly improved NLG capabilities. Large-scale language models, like GPT-3 and BERT, utilize transformer architectures to generate text. These models are trained on vast amounts of data and can capture complex language patterns, allowing them to generate coherent and contextually relevant text.

NLG models are usually fine-tuned on specific datasets for targeted tasks, such as text summarization, content creation, or chatbot responses. Some challenges in NLG include addressing issues of bias in the generated content, ensuring control over the output, and mitigating the problem of generating unrealistic or nonsensical text.

Ongoing research focuses on refining NLG models to be more interpretable, controllable, and capable of producing diverse and creative outputs. Additionally, efforts are being made to make NLG models more efficient to accommodate real-time and resource-constrained applications.

2.12 Conclusion

In this chapter, we've talked about the basics of machine learning and the different ways to

learn. Before this, we talked about neural networks, which are the most basic building blocks of deep learning. Then, we talked about GPT and the other deep-generating models being used to make text data. These background studies helped us do more work on our thesis.

Chapter 3

Literature Review

3.1 Introduction

Natural Language Generation (NLG) has emerged as a very dynamic and prolific area of study in recent years. Numerous highly intelligent individuals are actively engaging in collaborative efforts to develop cutting-edge technologies. Although several approaches for generation have been previously reported, their efficacy in delivering prompt and precise outcomes remains inadequate. This occurrence may be attributed to the existence of certain research deficiencies. This chapter provides an overview of several methodologies used in the field of language production, highlighting their respective contributions and identifying areas of study that need further investigation.

3.2 BERT: A Bidirectional Language Representation Model

Advanced language representation models have improved several NLP tasks. BERT—Bidirectional Encoder Representations from Transformers—is a major invention in this area. BERT emphasizes bidirectional contextual awareness, unlike Peters et al. (2018a) and Radford (2018). In this section, we explore BERT's unique design ideas and fundamental features to show how it creates solid language representations from unlabeled text[30].

BERT's basic idea is that reading material in both directions improves understanding. Earlier models focused on unidirectional situations. BERT innovates by conditioning all model layers on left and right context information. This bi-directional technique helps BERT under-

stand complicated semantics by capturing subtle linguistic linkages and nuances. This design decision impacts pretraining and fine-tuning, enabling state-of-the-art models with minimum task-specific adjustments.

BERT excels at many NLP tasks despite its simplicity. BERT's versatility makes it appealing. BERT's bidirectional context modeling design allows fine-tuning with one output layer. This simplifies BERT customization by reducing architectural changes. BERT's adaptability and resilience allow this pre-trained model to be used for question answering, language inference, and more.

3.2.1 BERT's Contribution

BERT has pioneered NLP via thorough examination, regularly outperforming benchmark workloads. This section showcases BERT's state-of-the-art outcomes in eleven NLP tasks. BERT's implementation improved the GLUE score by 7.7%, reaching 80.5±%. MultiNLI accuracy increased by 4.6±% to 86.7±%. In question-answering tasks, BERT scored 93.2 on SQuAD v1.1 Test F1, a 1.5-point increase. BERT improved SQuAD v2.0 Test F1 by 5.1 points to 83.1. These accomplishments demonstrate BERT's capacity to improve NLP tasks.

Task	Dataset	BERT Model	Accuracy
Natural language inference	GLUE benchmark	BERT-base	80.5
Question answering	SQuAD v1.1	BERT-base	93.2
Text classification	IMDB sentiment	BERT-base	91.9
Named entity recognition	CoNLL-2003	BERT-base	92.8

Table 3.1: BERT model performance on natural language processing tasks

3.2.2 BERT's Limitations

BERT is a strong language model with significant drawbacks. Training and running are computationally costly. BERT, a huge language model, demands a lot of computer resources to train and execute. This may hinder certain users, particularly those without strong computational resources. It struggles with lengthy texts. BERT processes sentences and paragraphs. Documents and publications may challenge it. BERT findings might be confusing. BERT is a black box model, making its predictions unclear. This makes BERT model debugging and improvement

challenging. It doesn't always adapt well. BERT is trained on a big text dataset but struggles to generalise to new jobs. If BERT is used for a job not well-represented in the training dataset, this might be an issue.

3.3 Attention Is All You Need

The development of dominant sequence transduction models has revolved around complex recurrent or convolutional neural networks, which consist of an encoder and a decoder. These models have achieved notable levels of success, particularly when enhanced with attention processes that establish connections between the encoder and decoder components. The interaction between these components facilitates enhanced comprehension of context in tasks involving sequence translation.

We, therefore, introduce the Transformer architecture, which represents a significant departure from the typical dependence on recurrent and convolutional architectures. This paradigm shift is considered pioneering in the field. The Transformer model, in contrast, is constructed solely based on attention mechanisms, thereby bringing about a significant transformation in the domain of sequence transduction. This unique methodology simplifies the construction of the network by eliminating the intricacies associated with recurrent dependencies and convolutional filters[31].

3.3.1 Advantages of attention mechanism

The superiority of the Transformer model in terms of quality, efficiency, and training speed is underscored by trials done on two independent machine translation jobs. In contrast to its previous iterations, the Transformer architecture exhibits improved parallelizability, facilitating accelerated training without compromising translation accuracy. The aforementioned efficiency leads to notable time savings, hence enabling the rapid creation and implementation of advanced translation models. The below table shows the performance transformer achieved initially.

Model	BLEU	Training Cost (FLOPs)	EN-DE	EN-FR
ByteNet	23.75			
Deep-Att + PosUnk	39.2	1.0×10^{20}		
GNMT + RL	24.6	2.3×10^{19}	1.4×10^{20}	
ConvS2S	25.16	9.6×10^{18}	1.5×10^{20}	
MoE	26.03	2.0×10^{19}	1.2×10^{20}	
Deep-Att + PosUnk Ensemble	40.4	8.0×10^{20}		
GNMT + RL Ensemble	26.30	1.8×10^{20}	1.1×10^{21}	
ConvS2S Ensemble	26.36	7.7×10^{19}	1.2×10^{21}	
Transformer (base model)	27.3	3.3×10^{18}		
Transformer (big)	28.4	2.3×10^{19}		

Table 3.2: Transformer Performance

3.3.2 Limitations

Attention processes may be confusing. Understanding how attention processes forecast is tricky. Debugging and improving attention models is tough. Attention is fragile. Attention models are sensitive to input data and perform poorly on data not well-represented in the training dataset. Slow attention processes. Long sequences need computationally costly attention methods. Real-time applications may be unsuitable.

3.4 Artificial intelligence versus Maya Angelou

The advent of publicly accessible and very efficient algorithms for natural language generation (NLG) has sparked considerable public interest and stimulated lively debates. The heightened focus on this phenomena may be attributed to the algorithms' purported capacity to produce written material that convincingly emulates human language in several topic areas. Nevertheless, in light of the prevailing excitement around this topic, it is important to acknowledge the dearth of empirical evidence, especially in relation to incentive activities. These activities would provide a comprehensive assessment of two crucial factors: the reliable distinction between text created by algorithms and material produced by people, and the genuine preference of readers

for one over the other[32].

Conducting Empirical Assessments In order to address this need, two separate but interrelated investigations were undertaken as a component of our research efforts. The aim of the study was to get a comprehensive understanding of the complexities associated with human reactions using GPT-2, a sophisticated algorithm designed for Natural Language Generation. The research used a sample of 830 individuals, carefully chosen to ensure representation of various demographic attributes. The experimental procedure included using initial lines of poetry that were generated by humans as a reference point. The GPT-2 model was then assigned the job of creating poetry extracts, which were then subjected to assessment using two independent methodologies. One methodology used was the random selection of a poem from the output of the algorithm, which was referred to as the ”Human-out-of-the-loop” strategy. In contrast, the ”Human-in-the-loop” approach included the inclusion of human interaction for the purpose of manually selecting the most appropriate poetry. Our analysis was based on the combination of algorithmically produced compositions with human-authored poetry.

3.4.1 Contribution

One notable aspect of our research involved the development of a novel iteration of the Turing Test, in which participants were assigned the responsibility of discerning between algorithmically-generated poems and those authored by human beings. The results were noteworthy, demonstrating a clear and discernible trend. The participants had significant difficulty in consistently distinguishing between poetry generated by algorithms and poems written by humans in the Human-in-the-loop scenario. On the other hand, the Human-out-of-the-loop condition demonstrated a greater degree of participant efficacy in discerning the poems that were generated by algorithms.

3.4.2 Limitations

The research was carried out using a limited number of participants. The research consisted of a rather small sample size of 40 individuals, which may limit the ability to establish conclusive findings. The research was carried out under the controlled environment of a laboratory. The outcomes of the study may have been influenced by the presence of a formal environment, perhaps leading participants to experience heightened pressure to provide certain replies. The

research did not account for the participants' pre-existing familiarity with algorithmic poetry. The participants' prior exposure to algorithmic poetry may have potentially affected their reactions, leading them to be more inclined to recognise it as such. The study did not assess the emotional reactions of the participants towards the poetry. This would have been beneficial in comprehending the reasons behind participants' adverse reactions towards algorithmic poetry.

3.5 Modern French poetry generation

In this study, we present a novel neural model that has been specifically developed for the generation of contemporary French poetry. The model we have developed is a combination of two pretrained neural networks, each designed to address distinct aspects of the process involved in generating poems. The fundamental structure consists of an encoder constructed based on the RoBERTa model and a decoder rooted in GPT-2. The integration of RoBERTa's sophisticated natural language comprehension abilities with GPT-2's proficiency in producing cohesive text enables us to exploit their respective strengths in a strategic manner[33].

The fundamental structure of our model consists of two basic components, namely the encoder and the decoder. The encoder, which draws inspiration from RoBERTa, demonstrates exceptional proficiency in extracting nuanced linguistic nuances, so establishing a robust basis for the future generation process. In the interim, the decoder, with GPT-2 as its foundation, converts these language representations into eloquent and contextually suitable poetic discourse. The utilization of a collaborative approach serves to reconcile the divide between comprehension and ingenuity, so empowering the model to construct poetic compositions that evoke a profound connection with their audience.

3.5.1 Contributions

RoBERTa and GPT-2 help the suggested model balance linguistic accuracy and creative ability. GPT-2 can create linguistically correct and aesthetically engaging material, whereas RoBERTa can comprehend minor language differences. The suggested approach generates contextually appropriate and aesthetically pleasing French poetry by merging these two models. Human judges assessed the model's effectiveness on a 5-point scale. The poem created scored 3.79 out of 5, the highest mark, for comprehensibility. The output's typicality and emotional resonance

scored 3.57 out of 5, the lowest score. Even the lowest score is excellent. These findings demonstrate the model's ability to effectively capture delicate language and create poetry that is intelligible and emotionally powerful. The model's ability to combine linguistic correctness with artistic expression suggests its broad applicability. For literary arts, creative writing, and language training, the approach might yield poetry. The model might also create music lyrics, scripts, and code.

3.5.2 Limitations

The model may have a predisposition for certain types of poetry or particular subject matters. The potential cause for this phenomenon may be attributed to the training data used in the model's training process. The model's capacity to develop poetry suitable for diverse audiences may be limited. The potential issue arises when using the algorithm to create poetry intended for youngsters or those with delicate subject matters. It is possible that the model's capacity to create poetry that adheres to factual accuracy may be limited. The use of the model for the purpose of generating poetry pertaining to historical events or scientific ideas may potentially give rise to an issue.

3.6 Bangla-BERT

The advent of pre-trained language models has signified a notable milestone in the domain of Natural Language Processing (NLP), heralding a novel epoch of sophisticated linguistic functionalities. Significantly, Transformer-based models, such as BERT, have received considerable attention because to their exceptional effectiveness in several language-related tasks. Nevertheless, it is worth noting that these models frequently demonstrate a notable inclination towards languages that possess abundant linguistic resources. This unintentionally results in the marginalization of other languages within the context of multilingual models, as shown by mBERT. The complexities associated with tackling this difficulty are further amplified when considering languages that encounter limitations in resources, such as Bangla. This paper explores the core challenges related to multilingual models and proposes a persuasive resolution in the shape of Bangla-BERT, a specialized monolingual BERT model designed specifically for the Bangla language[34].

Multilingual models refer to machine learning models that are designed to process and understand many languages. One of the primary challenges of multilingual models is the issue of data scarcity. Training a model to accurately comprehend and generate text in multiple languages requires a significant amount The utilization of mBERT for the purpose of accommodating many languages presents inherent challenges that are amplified when implemented for resource-limited languages such as Bangla. The limitations of mBERT's training dataset, which was designed to cover multiple languages, resulting in a diluted performance for specific languages. In addition, the inclusion of weights allocated to languages other than Bangla presents a barrier, diminishing the model's ability to concentrate on the distinctive intricacies of the Bangla language. Research conducted on several languages supports the notion that language-specific BERT models outperform multilingual models, emphasizing the necessity of dedicated models to fully harness the capabilities of unique languages.

3.6.1 Contribution

The study presents Bangla-BERT, a Bengali language model. Bangla-BERT, a version of the BERT (Bidirectional Encoder Representations from Transformers) paradigm, is successful for natural language processing applications. Bangla-BERT learns Bengali subtleties by pre-training on a huge Bangla text sample. Bangla-BERT outperforms multilingual models in binary language classification, multilabel extraction, and named entity identification. Bangla-BERT also outperforms Bangla fasttext and word2vec. Named entity recognition requires Bangla-BERT to consider sentence context. Bangla-BERT beats previous models on many benchmark datasets, according to the study. Bangla-BERT excels in several natural language processing tasks. The study also mentions Bangla-BERT's drawbacks, including as its ongoing development and high computing requirements. Bangla-BERT improves Bengali natural language processing, according to the study.

3.6.2 Limitations

The dataset used for training Bangla-BERT is somewhat limited in size. This may potentially limit the model's capacity to extrapolate to unfamiliar data. The Bangla-BERT model is characterised by high processing requirements for both training and execution. This may provide a potential obstacle for some users, particularly those who lack access to high-performance

computer resources. Bangla-BERT may be classified as a black box model, implying that comprehending the underlying mechanisms behind its predictions is challenging. Debugging and enhancing Bangla-BERT models might be challenging because to this issue. The evaluation of Bangla-BERT has been conducted on a restricted set of tasks. The extent to which it might effectively execute other tasks, such as machine translation or question answering, is uncertain as now. In general, the study conducted on Bangla-BERT exhibits encouraging outcomes. Nevertheless, it is essential to acknowledge and rectify some constraints that must be resolved before to the widespread use of Bangla-BERT.

3.7 Generating Classical Arabic Poetry

The evolution of poetry is a multifaceted endeavor that is intricately linked to the intricacies of meter and rhyme schemes. Historical undertakings within this specific subject often relied on thorough approaches for incorporating poetic aspects into mechanical systems. The integration of both meter and rhyme within the framework of poetry is a complex aspect that presents a difficulty for traditional computational methodologies. The complexity arises from the need to include not just the semantic essence of the text, but also the rhythmic and aesthetic components that define poetry. This study conducts an inquiry into a unique approach that leverages the functionalities of pre-trained language models, namely GPT-J and BERTShared. These models provide a thorough understanding of language patterns and structures, showing significant potential in identifying and reproducing the distinctive meters and rhyme patterns prevalent in ancient Arabic poetry[35]. The objective of this research is to assess the effectiveness of GPT-J and BERTShared models in understanding intricate themes present in poetry. Subsequently, we shall provide our empirical observations and results.

The act of producing poetry is not a routine computational procedure; instead, it entails grappling with the aesthetic intricacies that contribute to the uniqueness of each poem as a work of creativity. The fundamental nature of poetry is in the complex interaction of rhythmic patterns present in its lines, governed by established metrical norms, and the adept organization of rhyming vocabulary that enhances its aural appeal. Achieving a harmonious balance between language semantics and artistic structure is crucial when integrating these limits into automated systems. Historically, conventional tactics often used stringent approaches that heavily de-

pended on predetermined templates or carefully managed information, so limiting the potential for genuine creative expression. In addition, ancient Arabic poetry, which is well recognized for its intricate structures and intricate patterns of rhyme, poses a particularly formidable challenge. The intersection of semantic content and aesthetic form requires a new approach that successfully utilizes the capabilities of modern language models to decipher and replicate these intricate elements.

3.7.1 Contribution

Research shows that pre-trained language machines can compose classic Arabic poetry. This proves that AI can write coherent, contextually relevant poetry. The study highlights some of the obstacles to AI-generated Arabic poetry. Semantic consistency and plagiarism are these issues. Combining models may be the best way to generate Arabic poetry using AI, according to studies. GPT-J maintains rhythmic constancy, whereas BERTShared follows complex rhyme schemes. The findings may inspire new poem-generating methods. It might be used to construct poetry-generating or poetry-translating systems.

3.7.2 Limitations

For the purpose of the research, only a limited dataset of ancient Arabic poetry was employed. Because of this, it is possible that the models cannot generalize to other genres of ancient Arabic poetry or to current Arabic poetry. The capacity of the models to mimic the metrical and rhyming patterns seen in ancient Arabic poetry was the only criterion used to assess them in this research. The models were not evaluated based on how well they were able to capture the semantic content of the poetry. The research did not investigate how well the models performed in comparison to alternative approaches to writing ancient Arabic poetry, such as rule-based systems or poetry written by humans. Because of this, it is difficult to determine how well the models perform in comparison to other approaches.

3.8 Conclusion

This chapter has discussed a range of Natural Language Processing (NLP) articles that can be utilized for the generation of text data. We have provided a comprehensive analysis of the articles' capabilities, limitations, and our own empirical observations regarding their utility and efficacy. This chapter presents an overview of the latest advancements in NLP-based text data generation. These advancements have addressed the gaps left by previous studies and have significantly contributed to the progress of our own research work. In the subsequent chapter, we have introduced our exclusive approach for generating text data.

Chapter 4

Large Language Models

4.1 Introduction

Language is a fundamental and significant faculty possessed by individuals, enabling them to effectively convey thoughts and ideas, as well as engage in meaningful communication. This capacity begins to develop throughout the early stages of development and continues to progress and refine over one's whole lifespan. Machines, in their inherent state, lack the innate capacity to comprehend and engage in human language without the incorporation of advanced artificial intelligence (AI) algorithms. The endeavor to develop machines with the ability to comprehend, generate, and exchange information in a manner akin to human beings has posed a persistent and significant scientific obstacle. Language modeling (LM) is considered a prominent method for enhancing the linguistic capabilities of robots from a technical standpoint. In a broad sense, language modeling (LM) endeavors to represent the probabilistic likelihood of sequences of words, with the goal of forecasting the probabilities of forthcoming tokens or tokens that are absent. The research conducted by LM has garnered significant scholarly interest and may be categorized into four distinct phases of evolution.

4.1.1 Statistical language models

Statistical language models (SLMs) are computational models that utilize statistical techniques to capture the patterns and structures of natural language[36]. Statistical learning techniques (SLMs) were established during the 1990s as a foundation for the development of subsequent

SLMs. The fundamental concept involves constructing a word prediction model utilizing the Markov assumption, wherein the prediction of the subsequent word is made based on the immediate preceding context. The language models with a fixed context length n are commonly referred to as n-gram language models. Examples of such models include bigram and trigram language models. The utilization of SLMs has been extensively employed to improve the efficacy of task execution in the domains of information retrieval (IR) and natural language processing (NLP). Nevertheless, a common issue faced by these models is the curse of dimensionality. The proper estimation of high-order language models becomes difficult due to the exponential growth in the number of transition probabilities that must be calculated. Therefore, in order to address the issue of data sparsity, specific smoothing techniques such as backoff estimation and Good-Turing estimation have been proposed.

4.1.2 Neural linguistic models

Neural language models (NLMs) employ recurrent neural networks (RNNs) to assess the likelihood of word sequences[**?**]. The work made a significant contribution by introducing the concept of distributed representation of words and developing a word prediction function that relies on aggregated context characteristics, specifically the distributed word vectors. A unified solution for diverse natural language processing (NLP) applications was established by expanding upon the concept of acquiring efficient features for words or sentences through the utilization of a general neural network technique. Additionally, the concept of word2vec was introduced as a means to construct a reduced shallow neural network that can effectively learn distributed word representations. These representations have been shown to be highly effective in many natural language processing applications. These works have sparked the utilization of language models for the purpose of representation learning, extending beyond the scope of word sequence modeling. This development has had a significant influence on the field of natural language processing (NLP).

4.1.3 Pre-trained language models (PLMs)

The ELMo model was introduced as a means to capture word representations that are sensitive to context[37]. This is achieved through a two-step process: pre-training a bidirectional LSTM (biLSTM) network, followed by fine-tuning the biLSTM network for specific down-

stream tasks. Moreover, BERT was introduced as a result of pre-training bidirectional language models on extensive unlabeled corpora, utilizing the highly parallelizable Transformer architecture with self-attention mechanisms. These language models were trained with specifically designed pre-training tasks. The utilization of pre-trained context-aware word representations has proven to be highly beneficial in serving as general-purpose semantic features, significantly elevating the performance standards of natural language processing (NLP) activities. The findings of this study have served as a catalyst for a substantial body of subsequent research, establishing the ”pre-training and fine-tuning” approach to learning. In accordance with this framework, a considerable body of research has been conducted on pre-trained language models (PLMs), encompassing various architectural designs and enhanced pre-training methodologies. In this particular paradigm, there is often a need to carefully adjust the pre-trained language model (PLM) in order to effectively adapt it to various tasks that follow.

4.1.4 Large language models

Large language models (LLMs) are sophisticated computational systems that have been developed to process and generate human-like text. These models are characterized by their extensive size and complexity, enabling them to effectively understand and produce language-based content. The findings of the study indicate that the process of scaling PLM frequently results in an enhanced model capability when applied to subsequent tasks. Several research have been conducted to investigate the performance threshold through the training of progressively larger PLMs. The primary focus of scaling pertains to the enlargement of model size, while maintaining similar architectures and pre-training tasks. Notably, these larger-sized pre-trained language models (PLMs) exhibit distinct behaviors compared to their smaller counterparts, and demonstrate unexpected capabilities, commonly referred to as emergent skills, when addressing a range of intricate tasks. For instance, GPT-3 demonstrates the ability to effectively address few-shot tasks by employing in-context learning, a capability that GPT-2 lacks. Therefore, the scholarly community has coined the phrase ”large language models (LLM)” to refer to these PLMs of considerable size, which have garnered growing interest from researchers. One notable utilization of Language Models (LLMs) is demonstrated through the implementation of ChatGPT, which leverages the LLMs derived from the GPT series to facilitate discussion. This adaptation showcases an impressive capacity for engaging in conversations with human users. Following the introduction of ChatGPT, there has been a noticeable surge in the quantity of

arXiv papers pertaining to Language Models with Limited Memory (LLMs).

The current body of research extensively examines and investigates pre-trained language models (PLMs), however there is a noticeable lack of comprehensive reviews on language model fine-tuning techniques (LLMs). In order to provide motivation for our survey, we want to underscore three significant distinctions between LLMs and PLMs. Initially, it is noteworthy that larger language models (LLMs) exhibit certain unexpected emergent capabilities that may not have been noticed in their smaller predecessor models, known as previous smaller pre-trained language models (PLMs). The aforementioned qualities play a crucial role in enhancing the performance of language models while tackling intricate tasks, hence endowing AI algorithms with exceptional power and efficacy. Furthermore, the implementation of LLMs has the potential to significantly transform the processes involved in the development and utilization of AI algorithms by humans. In contrast to smaller PLMs, the primary method for accessing LLMs is typically through the utilization of a prompted interface, such as the GPT-4 API. It is imperative for individuals to have a comprehensive understanding of the functioning mechanisms of Language Model Models (LLMs) and to appropriately structure their jobs in a manner that aligns with the capabilities and requirements of LLMs. Furthermore, it is worth noting that the evolution of LLMs has blurred the traditional demarcation between research and engineering. The successful completion of LLM training necessitates a substantial amount of hands-on experience in the domains of large-scale data processing and distributed parallel training. In order to cultivate proficient LLMs, researchers must address intricate technical challenges, collaborating with or possessing expertise in the field of engineering.

4.2 Building blocks

Large language models (LLMs) often pertain to Transformer-based language models that encompass a substantial number of parameters, often in the range of hundreds of billions or even more. These models undergo training using extensive textual datasets. Some examples of Language Model Models (LLMs) are GPT-3, PaLM, Galactica, and LLaMA. LLMs possess the capability to comprehend natural language and effectively tackle intricate tasks by means of text production. In order to gain comprehension of the functioning of LLMs, this section furnishes foundational knowledge pertaining to LLMs, encompassing scaling rules, emerging capabili-

ties, and fundamental methodologies.

The present study investigates the scaling laws for liquid metal magnets (LLMs). The construction of LLMs involves the utilization of the Transformer architecture, which incorporates the stacking of multi-head attention layers within a deep neural network. These models exhibit an expansion in both model size and data size, as well as a substantial increase in total compute, resulting in enhanced model capacity. Two scaling laws have been postulated.

(a) **The KM Scaling Law:**

The KM Scaling Law, also known as the Karp-Miller Scaling Law[38], is a principle in computer science that relates the time complexity of a problem to the size of its input. The law proposed by Kaplan et al. elucidates the power-law correlation between model performance and the variables of model size (N), dataset size (D), and training compute (C). The scaling effect can be delineated by equations that elucidate the correlation between loss and the variables N, D, and C.

$$L(N) = \left(\frac{Nc}{N}\right)^{\alpha_N}, \quad \alpha_N \approx 0.076, \quad Nc \approx 8.8 \times 10^{13} \tag{4.1}$$

$$L(D) = \left(\frac{Dc}{D}\right)^{\alpha_D}, \quad \alpha_D \approx 0.095, \quad Dc \approx 5.4 \times 10^{13} \tag{4.2}$$

$$L(C) = \left(\frac{Cc}{C}\right)^{\alpha_C}, \quad \alpha_C \approx 0.050, \quad Cc \approx 3.1 \times 10^{8} \tag{4.3}$$

(b) **The Chinchilla Scaling Law**

The Chinchilla Scaling Law is a principle that describes the relationship between the size of an animal and many physiological and ecological factors. Hoffmann et al. proposed a novel scaling law that offers a computationally optimal training approach for Language Models (LLMs). The researchers conducted experiments involving different model sizes and data sizes, and then generated equations that enable the optimization of compute resource allocation.

$$L(N, D) = E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} \tag{4.4}$$

The present study aims to investigate the emergent abilities of Language Model Models (LLMs). The emergent abilities of large language models (LLMs) refer to capabilities that manifest in these models but are absent in smaller counterparts. These skills can be characterized as patterns in which performance exhibits a substantial improvement beyond random chance when the

model scale hits a specific threshold. This paper examines three emergent abilities.

$$Nopt(C) = G\left(\frac{C}{6}\right)^{a}, \quad Dopt(C) = G^{-1}\left(\frac{C}{6}\right)^{b} \tag{4.5}$$

(a) **In-Context Learning:**

The introduction of GPT-3 brought forth the capability to generate anticipated outcomes by leveraging natural language instructions and task examples. The aforementioned capability is observed in GPT-3 models of greater size, but it is absent in GPT-1 and GPT-2 models.

(b) **Proficiency in Instruction Comprehension**

Leveraging multi-task datasets, LLMs that have undergone fine-tuning have the ability to comprehend and adhere to task instructions for novel tasks, without relying on explicit examples. The capacity for generalization tends to enhance as the size of the model increases.

(c) **Sequential Reasoning:**

LLMs that employ the chain-of-thought prompting method have the ability to effectively address problems that require the execution of many reasoning stages. The manifestation of this capability becomes increasingly apparent as the size of the models increases.

Essential Techniques for Master of Laws (LLM) Students: Numerous strategies play a pivotal role in the achievement of LLMs.

The concept of scaling refers to the process of increasing or decreasing the size or magnitude of a certain phenomenon or system. Increased model/data volumes and enhanced training computations result in enhanced model capacity. Scaling laws provide a framework for effectively distributing computational resources.

(a) **Training:**

The topic of training will be discussed in this section. The utilization of distributed training techniques and optimization frameworks is imperative in light of the substantial size of the model. Techniques such as mixed precision training are known to enhance both the stability and performance of training processes.

(b) **Ability Elicitation:**

The process of designing appropriate task instructions and learning strategies within a

contextual framework to bring forth emergent abilities in language learning materials (LLMs).

(c) **Alignment Calibration:**

Language models with large-scale pre-training, such as LLMs, necessitate the process of aligning them with human values in order to mitigate the risk of generating deleterious material. Methods such as reinforcement learning with human feedback are employed to attain this alignment.

(d) **Manipulation of Tools:**

External tools have the ability to offset the lim:itations of LLM. The utilization of calculators or search engines has the potential to augment the capabilities of individuals pursuing a Master of Laws (LLM) degree.

The aforementioned methodologies and discoveries make valuable contributions to the advancement of LLMs, which have undergone significant progress in becoming proficient learners with broad competencies. The success of LLMs can be attributed to a range of variables, one of which being hardware advancements. This discourse centers on the primary technical methodologies and significant discoveries in the development of Language Learning Models (LLMs).

4.3 Types of language models

Certainly, let's delve into more detail about each type of large language model based on the transformer architecture:

4.3.1 Autoregressive

Autoregressive models operate on the principle of generating text one word at a time, where the prediction of each word depends on the preceding words in the sequence[39]. These models employ a left-to-right generation approach, mimicking the way humans construct sentences. The training process involves maximizing the likelihood of the next word given the previous

context. GPT, including versions like GPT-3 and GPT-4, is a prominent example of this category. GPT models excel at coherent text generation and are often utilized for creative writing, content generation, and conversational agents. They have the ability to understand context and produce contextually relevant responses.

4.3.2 Autoencoding

Autoencoding models, in contrast, focus on learning representations of text by transforming input sentences into a compact form known as embeddings. BERT, a groundbreaking example, employs a bidirectional approach to learn contextual information from both directions in a sentence. During training, BERT learns to predict missing or masked words in the input text, thus capturing contextual nuances. BERT's pre-trained representations can then be fine-tuned for specific NLP tasks, making it a versatile choice for tasks like sentiment analysis, question answering, and more[40].

4.3.3 Combined Models

Combined models merge the best of both worlds. The Text-to-Text Transfer Transformer (T5) is a prime illustration of this approach. T5 reformulates a wide array of NLP tasks as text-to-text problems. This transformation allows the model to handle diverse tasks using a unified framework, where inputs and outputs are treated as textual sequences. By utilizing a shared architecture, T5 achieves strong results across tasks like translation, summarization, and classification. This versatility streamlines model deployment, reduces the need for specialized architectures, and enhances performance across a broad range of tasks.

In summary, large language models based on the transformer architecture are revolutionizing the field of natural language processing. Autoregressive models like GPT excel in creative text generation, autoencoding models like BERT master contextual understanding, and combined models like T5 offer unified frameworks for various NLP tasks. These models represent a significant step toward AI systems that can understand and generate human-like text, with implications spanning from content creation to automated customer service and beyond.

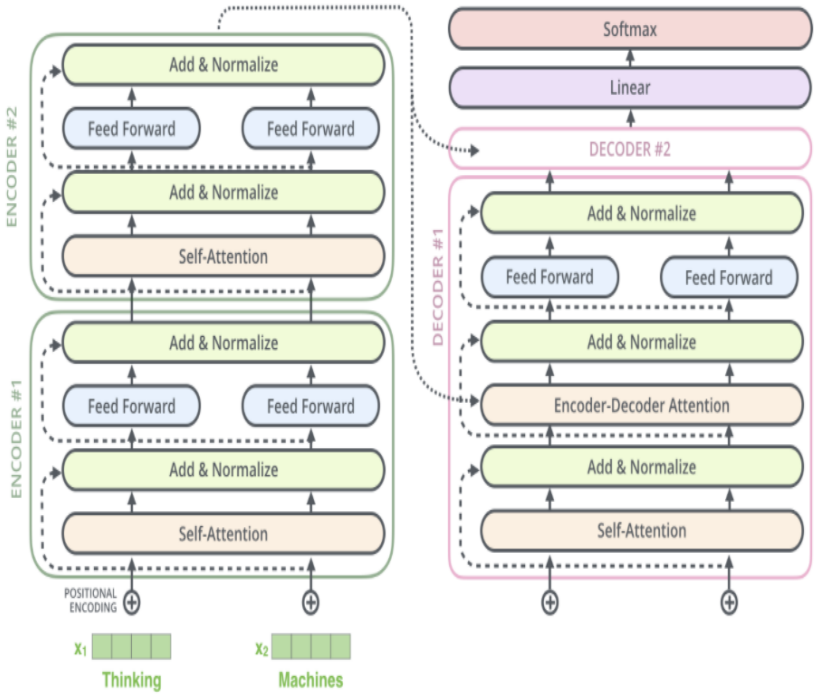


Figure 4.1: T5 Architecture

4.4 Conclusion

In this chapter, we have covered a variety of language models that may be used to generate text data. We have detailed the models' capabilities, shortcomings, and our own observations on their usefulness and abilities. This chapter pictures the recent works of generation of text data and researches for Bangla language, which allowed us to fill in the blanks left by earlier studies and advance our own research work. In the following chapter we have presented a methodology for implementation.

Chapter 5

Methodology and Implementation

5.1 Introduction

This chapter explores the complex procedure of creating Bangla text through the utilization of Natural Language Processing (NLP) techniques. The method commences with the initial preprocessing and tokenization stages, which are specifically designed to handle the distinctive linguistic characteristics of the Bangla language. The utilization of language modeling incorporating attention mechanisms has been shown to significantly improve comprehension and fluency. Transfer learning from pre-trained models and the utilization of fine-tuning the model with custom dataset are effective strategies for mitigating the issues associated with a scarcity of labeled data. The text acknowledges the presence of problems, namely those related to morphological complexity and cultural factors. The ethical ramifications of bias and sensitivity are given significant emphasis. In conclusion, the integration of linguistics and natural language processing (NLP) facilitates the production of culturally appropriate and logically consistent Bangla literature, hence promoting effective communication in the language.

5.2 Overview of the Proposed System

The system that we have proposed primarily consists of the following eight components. : data collection, data preprocessing, tokenization, encoding, model fine-tuning, decoding, generating texts, and evaluating texts. The primary constituents encompass certain subordinate elements as well. In the subsequent sections, a comprehensive analysis of each block is provided.

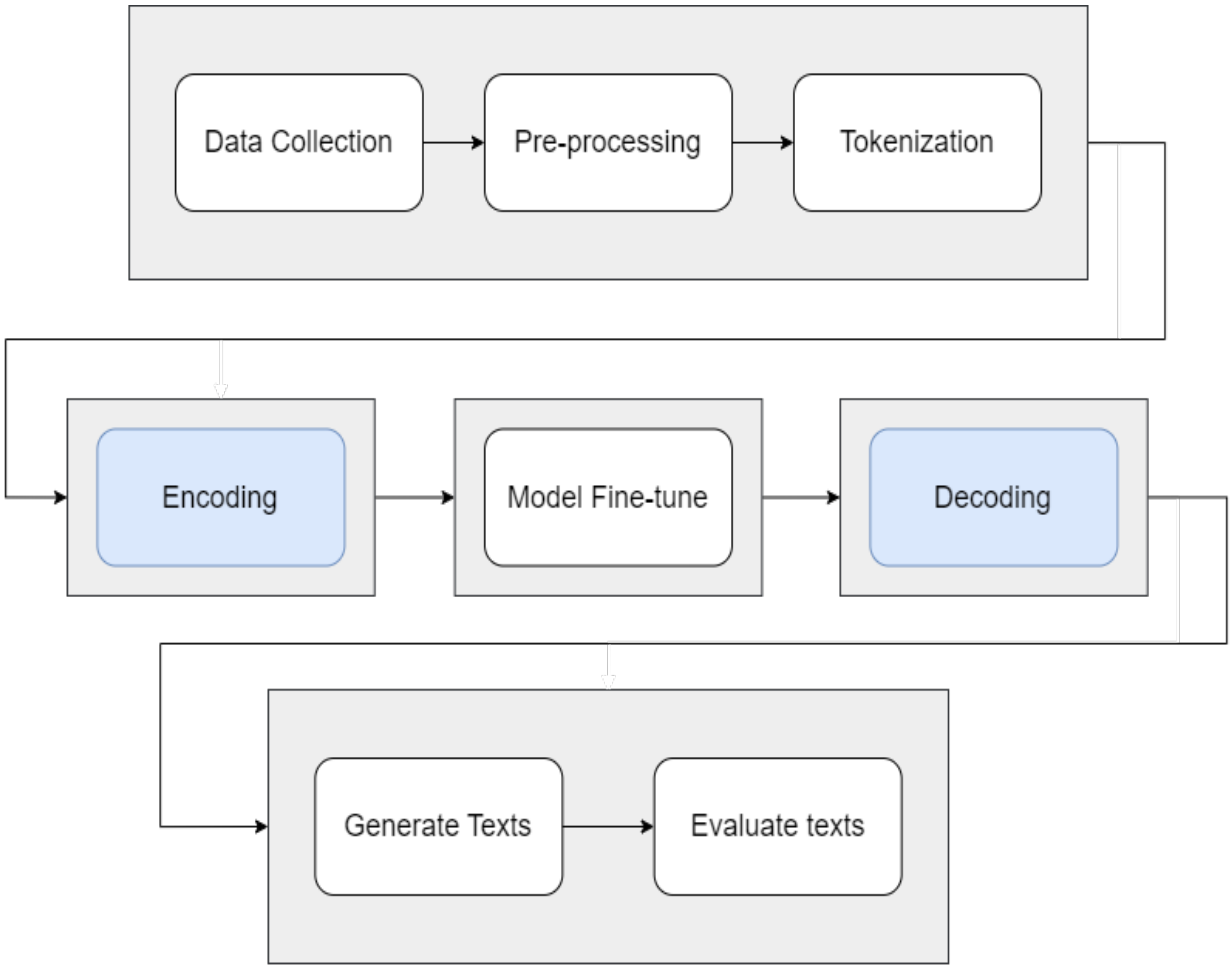


Figure 5.1: Overview of the Proposed System

5.3 Data Collection & Description

The practice of ”web scraping” has emerged as an essential method in the field of archiving priceless works of literature. In order to accomplish this goal, the ageless novels of ’Feluda,’ which were fashioned by the literary master Satyajit Ray, have been painstakingly retrieved. This collection is comprised of a total of ten novels, which together form a web that is comprised of 165,708 words and 890,586 characters[41]. The ten books in the dataset are- multirow

Used for	Book
Train	Gosaipur Sargaram
	Ghurghutiar Ghotona
	Gangtok E Gondogol
	Baksho Rahasya
	Feludar Goyendagiri
	Amber Sen Ontordhan Rohosso
	Chinnomastar Obhishap
	Badshahi Angti
	Gorosthane Sabdhan
Test	Bombay er Bombete

The relevance of this undertaking extends further than the simple collection of data. This dataset has been meticulously curated in order to make it easier to adjust the parameters of language models. A crucial part is played by 'Bombay er Bombete' in a noteworthy validation dataset. The fusion of technology and literature is a demonstration of how seriously we take our responsibility to protect our cultural and literary legacy. This effort assures that the legacy of 'Feluda' will continue to live on and may be accessed by the public by drawing upon the power of web scraping. The ongoing impact of Satyajit Ray's detective series is highlighted by the fact that these efforts encapsulate the marriage of modern methods with traditional texts. As the process of preservation moves forward, it serves as a monument to the literary skill of one of the most talented storytellers of our era and stands as a testament to that skill.

5.4 Data Pre-processing

Data preparation is crucial to data analysis and machine learning[42]. Clean, process, and organize raw data for analysis or model training. Handling missing data, outliers, standardizing or normalizing features, and encoding categorical variables are part of this procedure. Effective data preparation improves data insights and machine learning model performance by providing correct and relevant data. The most significant text data preparation methods in natural language processing (NLP):

1. **Eliminating the Use of Punctuation**: It's true that punctuation marks like periods, commas, and question marks help with grammar, but it's also possible that they don't contribute all that much to the meaning of the text. Eliminating punctuation marks from a text makes it simpler, which in turn makes it simpler for algorithms to concentrate on the important parts of the text. This is especially essential in activities such as sentiment analysis, which can discern the tone of a statement even when the punctuation is absent.

2. **Delete the following stopwords:** Stopwords are words that are used frequently in a language but do not typically contain any significant meaning. Some examples of stopwords include ”and,” ”the,” ”is,” and ”in.” The noise in the dataset can be reduced and the processing can be sped up by removing stopwords. By removing these terms, the algorithms will be able to concentrate on the words that have a higher semantic value, which will result in greater accuracy when doing NLP tasks.

3. **The process of stemming and lemmatization:** The process of stemming entails stripping words of their prefixes and suffixes in order to reveal their original root form. As an illustration, the word ”running” would be shortened to ”run.” This is taken a step further by lemmatization, which is the process of transforming words to their dictionary or basic form. This improves the model's comprehension of text as well as the coherence of the dataset by ensuring that different forms of a word are considered as equivalents to one another and enhancing the coherence of the dataset[43].

4. **Checking for Misspellings and Making Corrections:** Finding and fixing typographical problems in a piece of writing is what's meant by ”checking and correcting” the spelling of words. This phase improves the quality of the data and helps to avoid misunderstandings caused by mistakes in the text. Correct spelling makes it more likely that the model will correctly grasp the meaning that was intended to be conveyed by the text, which ultimately results in more accurate analysis.

5. **Getting rid of URLs and characters with special meaning:** Text obtained from websites may contain URLs, characters not commonly used on the web, and other symbols that are irrelevant to the research. The removal of these components will result in a reduction in background noise and will guarantee that the attention is maintained on the textual content. This is vital for ensuring the data's integrity while also enhancing the NLP algorithms' performance.

6. **Dealing with Abbreviations and Acronyms:** Increasing the usefulness of abbreviations and acronyms requires transforming their shorthand forms into their complete formulations. Because of this, the model will always have an accurate understanding of the context and will retain a consistent interpretation of the text. The accuracy and coherence of the analysis are not affected when abbreviations are expanded[44].

5.5 Tokenization

Tokenization is the fundamental building block of natural language processing (NLP)[45], which deciphers the complexities of the human language and gives computers the ability to read, analyze, and synthesize text in an efficient manner. We delve into the fundamental relevance of tokenization in this investigation, as well as its methodology, problems, and vital role in bridging the gap between linguistic diversity and computational precision.

Tokenization is significant because it can break down human language, which is a continuum of sounds and meanings, into discrete units that machines can understand. This ability gives tokenization its significance. This transformation is essential for a variety of tasks involving natural language processing (NLP), such as text categorization, sentiment analysis, machine translation, and many more[46].

Take the following line under consideration: ”She enjoys hiking in the mountains.” Through the process of tokenization, it is partitioned into a number of distinct tokens, such as [”She enjoys hiking in the mountains”]. Each token has its own unique semantic value, and by segmenting the data in this way, computers are able to process each token independently, thereby gaining context and developing insights.

Tokenization Methods:

5.5.1 Word-Level Tokenization:

Word-level tokenization is a technique used to segment text into discrete units, namely individual words. This approach considers individual words as discrete units, making it particularly suitable for tasks that need a comprehensive understanding of the text under analysis.

5.5.2 Subword Tokenization:

This process divides the text into smaller components, such as syllables or morphemes, so that it can be more easily analyzed. This method is especially helpful for languages that have intricate morphological structures, as well as for applications like sentiment analysis that demand a more nuanced comprehension of text, such as when used.

5.5.3 Byte-Pair Encoding:

BPE is a method of subword tokenization that finds character sequences that occur frequently and encodes them as a single token. This method takes into account uncommon terms and increases the effectiveness of vocabulary representation.

5.5.4 Tokenization at the Sentence Level:

Text can be broken up into sentences using a process called sentence-level tokenization. It is necessary for activities such as automatic translation and summarization, in which sentences serve as the fundamental building blocks of the analysis.

The cornerstone of natural language processing, known as tokenization, helps to close the gap between human language and computational analysis. Its central function of parsing text into comprehensible chunks is the fundamental building block for both the understanding and production of language by robots. Tokenization is continuously developing alongside natural language processing (NLP), which allows it to accommodate linguistic variation while also enhancing efficiency and the quality of our interactions with robots that can understand and communicate in human language.

5.6 Encoding

Encoding in natural language processing (NLP) is essential for machine learning models to understand textual data. This lets machines understand human language's semantics and context[47]. Encoding, its procedures, and its importance in text analysis are examined in this study.

Encoding converts human-readable text into machine learning-friendly numerical representations. These numerical representations capture text semantics, allowing machines to analyze, classify, produce, and translate the language.

Machines struggle to understand human language's complexities. Encoding converts words, phrases, and sentences into numerical vectors, making algorithms more efficient. Some encoding methods are:

5.6.1 One-Hot Encoding:

Each word is a binary vector with a "1" in the vocabulary index position and "0"s elsewhere. This straightforward method produces sparse, high-dimensional vectors.

5.6.2 Word embeddings:

Word embeddings are dense, continuous vectors in lower dimensions. Word2Vec, FastText, and GloVe capture semantic links between words, increasing model context and meaning.

5.6.3 Subword Encoding:

BPE and SentencePiece tokenize words into subwords or characters. This method handles unusual words and morphological variants, essential in complicated languages.

5.6.4 Positional Encoding:

BERT and GPT use positional encodings to capture word order. These encodings improve contextual understanding in word embeddings[48].

NLP encoding transforms text into machine-readable representations. This approach connects human language complexity and machine learning model capabilities. Encoding allows machines to interpret, understand, and synthesize text, revolutionizing communication, analysis, and production in NLP applications.

5.7 GPT-2 fine-tuning

The process of fine-tuning GPT-2 entails modifying the pretrained model to suit a particular job or domain. Although a comprehensive mathematical derivation is beyond the scope of this response, I will present a high-level summary of the phases involved in the fine-tuning process and discuss the fundamental notions that underpin it. The following steps outline the process of fine-tuning:

5.7.1 Data Collection and Preprocessing:

Acquire and preprocess a dataset that is pertinent to the designated activity. It is recommended that the dataset be appropriately labeled or annotated to accurately represent the desired outcome. For instance, in the context of fine-tuning for text classification, it is necessary to have a dataset that consists of input text paired with their respective labels[49].

5.7.2 Selection of the Model:

I had selected the GPT-2 variation which was pre-trained on bnc4 Bengali dataset that aligns with the appropriate scale and intricacy required for this specific undertaking[50]. The choice of GPT-2 model size is contingent upon the available resources and the level of complexity associated with the given task.

5.7.3 Tokenization:

The input text is tokenized and each token is mapped to its respective token ID in the GPT-2 vocabulary. The aforementioned procedure is crucial for the model to effectively analyze the input data.

5.7.4 Description of Model Architecture and Parameters:

The GPT-2 model is initialized by employing its pretrained weights. GPT-2 is constructed using the transformer architecture, which incorporates numerous layers of self-attention and feedforward neural networks. In the process of fine-tuning, it is common practice to incorporate a task-specific output layer onto the GPT-2 base model.

The loss function is a mathematical function used in machine learning algorithms to quantify the discrepancy between predicted and actual values. A loss function is a mathematical construct used to quantify the dissimilarity between the predictions made by a model and the true labels associated with the data. The selection of the loss function is contingent upon the particular task at hand. An instance where categorical cross-entropy loss could be employed is in the context of text classification.

5.7.5 Training:

The model should be initialized with pretrained weights and subsequently fine-tuned using a dataset suitable to the job at hand. During the training process, the model's parameters are updated through the utilization of gradient descent and backpropagation techniques. The model acquires the ability to provide predictions that are tailored to specific tasks, while still preserving the knowledge obtained from the pretrained GPT-2 model.

Fundamentally, the process of fine-tuning entails the adjustment of the model's parameters, specifically the weights and biases, with the objective of minimizing the designated loss function. The aforementioned procedure is commonly executed via gradient descent optimization. The computation of the gradients of the loss with respect to the model parameters is achieved by the utilization of backpropagation. Subsequently, the model parameters undergo iterative updates in order to minimize the loss.

The process of fine-tuning in the GPT-2 model architecture entails the computation of gradients for the task-specific output layer, followed by the propagation of these gradients through the layers of self-attention and feedforward networks via backpropagation. The aforementioned procedure involves modifying the parameters of the overall model in order to enhance its performance in relation to the given task[51].

The mathematical calculations utilized in the fine-tuning of GPT-2 can be intricate, owing to the inherent complexity of the transformer architecture and the backpropagation procedure.

5.8 Decoding

Decoding in natural language processing (NLP) converts numerical computer predictions into human-readable text. Language generation, machine translation, and text summarization require it. Decoding allows machines and humans to communicate. Decoding, its methods, problems, and importance in making NLP models accessible and impactful are examined in this study[**?**].

Decoding opposes encoding. Decoding turns numerical representations of text into intelligible, grammatically correct human language. Communicating NLP model insights and generated information to users requires this approach.

Decoding Methods:

5.8.1 Greedy Decoding:

Greedy decoding selects the highest-probability token at each generation step. This easy method may result in repetitious or strange text.

5.8.2 Beam Search:

Beam search keeps the top-k tokens from greedy decoding. It investigates ways to improve the generated text.

5.8.3 Top-k sampling:

The model randomly samples the top-k most likely tokens at each stage, diversifying the resulting text.

5.8.4 Nucleus Sampling:

Nucleus sampling, like top-k sampling, focuses on the top-p probability mass of the distribution, enabling more text variety.

5.8.5 Temperature scaling

It changes the softmax distribution used for sampling, controlling output unpredictability. Temperature increases unpredictability.

NLP model predictions are understood by humans after decoding. The technique makes model results accessible, engaging, and meaningful for consumers across many applications. As NLP advances, decoding methods shape how machines connect with humans and enable more natural, coherent, and interactive language generation.

5.9 Generating Texts

The fundamental components of AI language models are known as prompts and text production. The process is kicked off by a prompt, which guides models to produce language that is coherent[52]. Models like GPT-3 are particularly effective at this, producing contextually rich information in response to a wide variety of inputs. This dynamic pair is valuable in a variety of contexts, ranging from helping with coding to creative writing. There are still many obstacles to overcome, such as assuring accuracy and preventing prejudice. There is much less training data available for other languages. This means that applications like ChatGPT(trained by GPT-3) may not be able to learn the nuances of those languages as well as they can learn English. That is why in another language besides English it fails the Turing test or respective evaluation[53]. Maintaining a healthy equilibrium between control and creativity is essential. This synergy opens the door to human-AI interaction, in which computers react and produce depending on our inputs, transforming the way humans interact with AI in the process.

5.10 Evaluating Texts

In the field of natural language processing (NLP), one of the most typical tasks is to compare and contrast the similarities and differences between two different texts[50]. The degree of resemblance between two pieces of text can be determined using a number of distinct approaches, some of which are more straightforward than others. The following are some possible approaches:

1. **Cosine Similarity:** This is a standard approach that is used to evaluate the degree to

which two documents may be represented as vectors in a space with a high dimension. When representing documents with methods such as TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings, it is frequently utilised. Cosine similarity is a measurement that compares two vectors based on the cosine of the angle that separates them.

2. **The Reading Ease score:** It indicates how simple it is to read and comprehend a piece of text. Several formulas and scales can be used to calculate the Reading Ease score, with the Flesch-Kincaid Readability Formula and the Flesch Reading Ease Formula being the most well-known. Both of these formulas consider sentence length and syllable count when determining the legibility of a text.

   The grade generated by the Flesch Reading Ease Formula ranges from 0 to 100. Text with a higher score is simpler to read, while text with a lower score is more difficult to comprehend. Here is a general interpretation guideline for the scores:

   Keep in mind that these are approximate scores that can differ depending on the specific text and the formula used. Similar texts may receive marginally different scores when calculated using distinct formulas. Additionally, topic complexity and the reader's familiarity with the subject matter can impact how a text is perceived in terms of intelligibility.

3. **Human evaluation:** In this process the generated literature is evaluated by a human observer. It provides a more nuanced and complete text quality evaluation than automated measures, making it the gold standard for assessing natural language-generating systems.

   Evaluators may also compare produced and human-authored content. Asking assessors to distinguish machine-generated from human-written content does this. Evaluators may also score the texts' resemblance. Evaluation objectives determine evaluation approach. A rating scale may be adequate to evaluate produced text quality. Identifying particular areas for development may need a more qualitative approach.

5.11 Implementation

The model was implemented on Google Colab using Nvidia T4 GPU. The NVIDIA T4 GPU was unveiled in 2018 as a Turing-based GPU. It incorporates multi-precision Turing Tensor

Cores and brand-new RT Cores and is designed for mainstream computing environments. The T4 GPU has 16GB of GDDR6 memory and 7.8 TFLOPS of peak performance. In addition, it contains 896 CUDA cores and 40 RT cores. The model and tokenizer was taken from hugging face[54] where the model was pre-trained with mc4 Bengali dataset[55] . The Model parameters size was 124.4 Million.

The training arguments were:

Argument	Value
Evaluation strategy	Epoch
Number of epochs	35
Per device train batch size	16
Per device evaluation batch size	32
Evaluation steps	80
Save steps	5000
Weight decay	0.01

Table 5.1: Training Arguments

5.11.1 Model Architecture Summary

Argument	Value
Word embedding dimension	768
Positional embedding dimension	768
Number of attention blocks	10
Hidden dimension	768
Activation function	GELU
Dropout rate	0.01
Bias	False

Table 5.2: Hyperparameter List

5.11.2 Evaluating function

The process of refining or adjusting The GPT language model, known for its sophistication, undergoes performance evaluation throughout the training process to achieve optimum outcomes. The aforementioned assessment function plays a crucial role in providing guidance throughout the fine-tuning phase. Periodically, the output produced by the model is evaluated against the ground truth data, enabling a quantitative evaluation of its predicted accuracy. The negative log-likelihood loss[56] is given by:

$$L = -\log P(t \mid T)$$

1. The term ”L” denotes the negative log-likelihood loss function. The metric assesses the degree of alignment between the anticipated probability distribution of the model and the actual probability distribution of the target token.

2. The term ”t” refers to the subsequent token in the sequence that one aims to forecast.

3. ’T’ is the representation that denotes the sequence of tokens that come before. Within the framework of a language model, the contextual or historical information used by the model serves as the basis for predicting the subsequent token.

4. The expression P(t | T) represents the probability assigned by the model to the occurrence of the next token ’t’, given a sequence of previous tokens ’T’. Fundamentally, the model provides a forecast for the likelihood of the subsequent token based on contextual information.

5. The logarithm function, denoted as ”log,” in this context refers only to the natural logarithm function.

   The negative log-likelihood loss function is used as an evaluation metric to assess the degree of alignment between a model’s predictions and the observed data. The act of reducing this loss throughout the training process aids the model in acquiring the ability to make predictions of sequences that have a higher probability of occurrence.

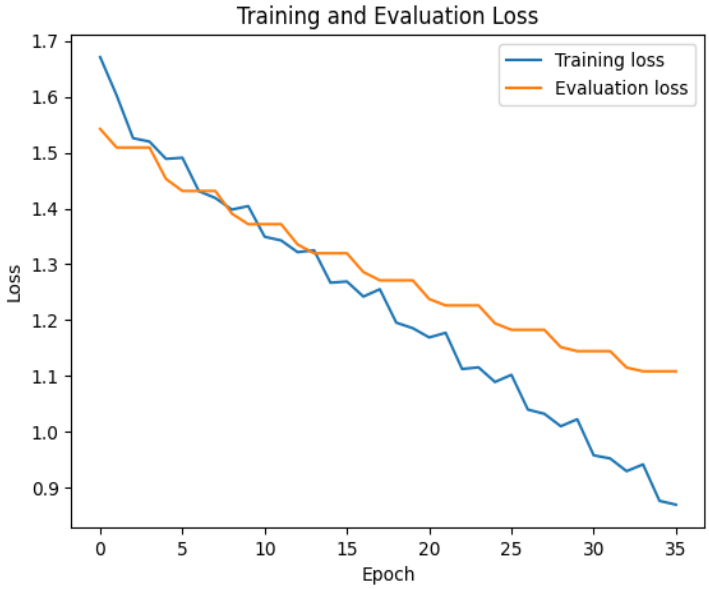


Figure 5.2: Training and evaluation loss during fine-tuning

5.12 Conclusion

In this chapter, a comprehensive outline of our thesis work has been presented. We have covered data collection, data analysis, and data preparation, amongst other topics. Then, we have explained the Gpt-2 model and other language models. Finally, we have discussed our generation method and the environment we used for implementation. The following chapter consists the results of these processes we have mentioned in this chapter.

Chapter 6

Result and Performance Analysis

6.1 Introduction

Natural Language Generation (NLG) is indispensable for transforming structured data into human-readable text. It is essential to evaluate the results and efficacy of NLG systems in order to determine their effectiveness. This analysis examines the contributions of three distinct metrics - Reading Ease Score, Cosine Similarity, and Human Evaluation - to this evaluation.

The Reading Ease Score is a measurement of the text's intelligibility and simplicity. It uses formulas such as Flesch-Kincaid or Flesch Reading Ease to designate a score indicating the text's level of difficulty. Higher scores indicate text that is simpler to comprehend, which is advantageous for diverse audiences. A low score may indicate that the NLG algorithms must be modified to improve text accessibility. In a vector space, Cosine Similarity quantifies the semantic similarity between generated and reference text. This metric measures the similarity of meaning between two texts. Higher cosine similarity values indicate that the NLG system is producing content that semantically aligns with the intended output. However, it may not account for the originality and diversity of generated content.

Human evaluation remains an essential component of NLG evaluation. Human evaluators provide nuanced insights that automated metrics may overlook, such as context-appropriateness, tone, and creativity. Human feedback collected through studies or expert evaluations provides a complete comprehension of NLG performance. It identifies issues that metrics may overlook, such as text that is technically accurate but sounds unnatural.

6.2 Flesch Reading Ease Score

The Reading Ease Score is a mathematical formula used to evaluate the level of readability in a given text, considering many parameters like sentence length, word count, and syllable complexity. The objective of this scoring system is to provide an approximation of the level of comprehension required to grasp a given text. The term is often used in many settings, including instructional resources, technical books, and literary works[57].
The Flesch Reading Ease Score formula is derived from three primary components:

1. The average sentence length refers to the mean amount of words included in each sentence. The complexity of a text may be increased by using longer sentences, which may therefore make the content more challenging to comprehend.

2. The metric being discussed here is the average number of syllables found in each word. Words that include a greater number of syllables tend to provide a higher level of difficulty in terms of reading and comprehension.

3. Character per Word (CPW) is a metric that quantifies the average amount of characters included inside each word. The use of lengthier vocabulary choices might enhance the general intricacy of the written discourse.

The formula by which the score of a text is calculated is given below.

$$Score = 206.835 - (1.015 \times \frac{\text{words}}{\text{sentence}}) - (84.6 \times \frac{\text{characters}}{\text{word}})$$

For calculating the scores, we first translated the generated texts into English. After that, each of the generated texts from the model was calculated. It is worth mentioning that, texts generated by ChatGPT were not part of this score evaluation.

Score Range	Readability Description	Texts
0-30	Extremely difficult to read	0
30-50	Difficult to read	0
50-60	Fairly difficult to read	4
60-70	Plain English	7
70-80	Fairly easy to read	15
80-90	Conversational English	12
90-100	Very easy to read	2

Table 6.1: Readability Descriptions

For evaluating the texts, 40 sample texts were generated from the fine-tuned GPT-2 model Measuring Readability by translating the texts in English. Flesch Reading Ease score was used which ranges from 0 to 100.[58]

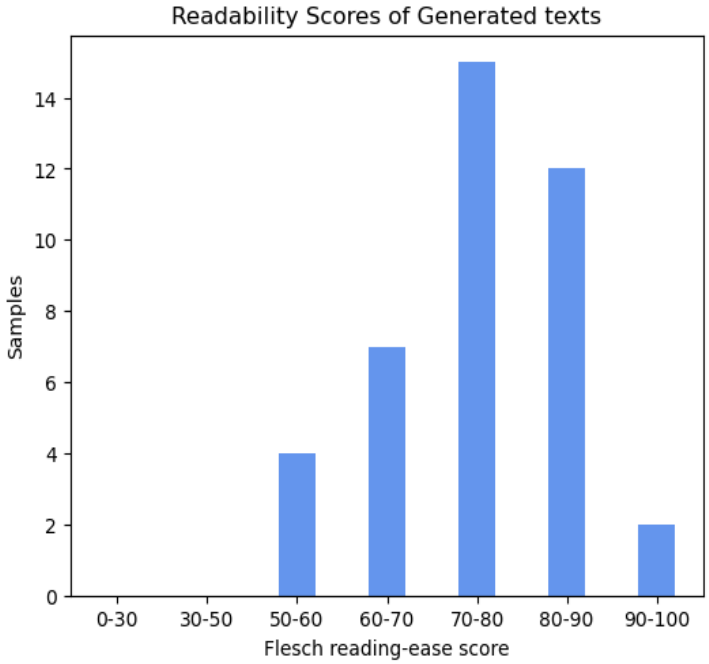


Figure 6.1: Readability Scores in each range

6.3 Human Evaluation

20 texts were generated from the fine-tuned model from a list of promts. The same prompts were given to ChatGPT and it was asked to generate texts in the style of Satyajit Ray. The GPT-2 generated texts were a little better than ChatGPT responses to the human reviewers who participated in the evaluation,[59]. A Google form was created to conduct the survey and 23 responses were recorded. 40 texts were generated using the same input used in the model. In one form, a mixture of 20 ChatGPT generated and 20 human generated texts were provided.

Category	Count
Predicted ChatGPT output as Human written	105
Predicted ChatGPT output as AI written	21
Predicted Model output as Human written	743
Predicted Model output as AI written	51
Successful Prediction	80.76%

Table 6.2: Prediction Results

Prediction Statistics

Total Predictions = $23 \times 40 = 920$

Prediction Type	**Count**
Human Written	743
Successful Prediction	$\left(\frac{743}{920}\right) \times 100$

Table 6.3: Prediction Statistics

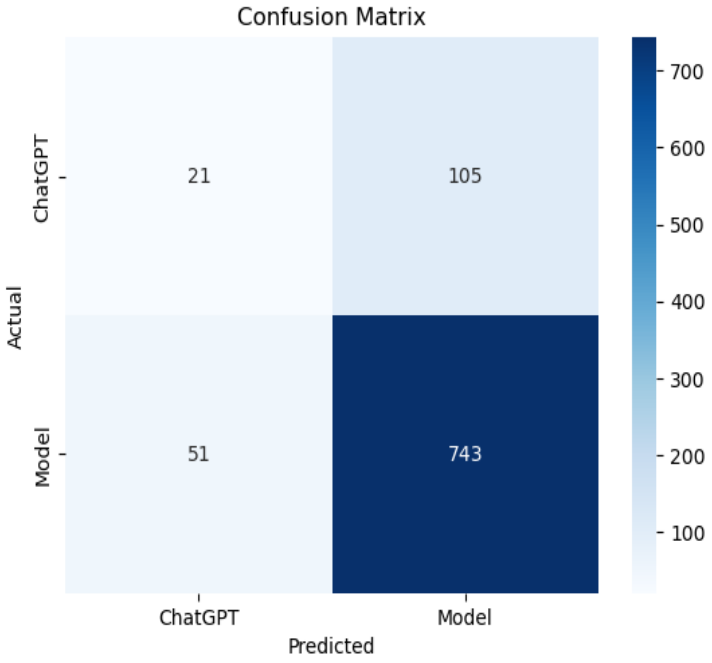


Figure 6.2: Confusion matrix of human prediction and actual text nature

6.4 Performance Analysis

The content generated by our methodology had a commendable level of cohesiveness inside each respective paragraph. The feasibility of this achievement was facilitated by the inherent coherence of the dataset used for the process of fine-tuning. The loss could be even less if we had used a labeled dataset for fine-tuning instead of relying on context length-based input-output labeling of the text corpus[60]. In some instances, human evaluators were deceived by the inherent consistency of the generated text, leading them to believe that it was authored by people, even when compared to responses generated by ChatGPT. Although ChatGPT is based on the architecture of GPT-3.5, a notable distinction lies in the absence of the specific dataset used in our work throughout both the pre-training and fine-tuning phases. Indeed, it is accurate to state that ChatGPT employs the underlying framework of GPT-3.5. Significantly, the inherent capacity of GPT-3.5 enabled it to generate sentences that adhere to proper grammatical rules. However, it became evident that the material it created consistently surpassed expectations in terms of overall coherence. The observed discrepancy may be attributed to the distinctive characteristics of the fine-tuning dataset, which inherently modified the level of contextual depth

and semantic coherence in our generated content. This ultimately led to the enhanced coherence and nuanced significance of the generated material.

6.5 Conclusion

During the course of this chapter, we explored several facets pertaining to language modeling and assessment. Furthermore, we conducted an analysis of the Flesch Reading Ease Score, which serves as a significant instrument for evaluating the readability of text that has been created. The method takes into account factors such as sentence length, syllables per word, and characters per word, in order to provide a quantitative assessment of the complexity of a given text. Through the use of these approaches, it is possible to optimize models in a very effective manner and assess the comprehensibility of their outputs. The use of this complete strategy serves to improve both the quality and accessibility of natural language-generating systems.

Chapter 7

Conclusion and Future Works

7.1 Introduction

Understanding the subject and history typically initiated the research process. This was followed by a literature review, an explanation of GPT-3, an overview of the methodology, a presentation of the empirical findings, and a summary of the study journey. We scrutinized the factors, events, and influences that shaped the field. This process assisted in clarifying the research issue and uncovering gaps and contradictions. Scientific, academic, and research papers were examined. This aided in determining the field's current state and identifying knowledge gaps. Large textual databases trained these computers to emulate human language. This illuminated their architecture, training methods, and capabilities for natural language understanding and generation.

In the chapter that came before this one, both the results of our experiments and an analysis of how well they performed were discussed. This chapter contains a summary of all of the research that we have done, followed by a discussion of the limitations of our investigation. After this, there is a discussion of possible works to be produced in the future, and then the chapter comes to a close.

7.2 Thesis Summary

The research inquiry commences with an analysis of the historical context and fundamental knowledge that provide the framework for the issue being investigated. This encompasses a thorough examination of the various elements, events, and influences that have contributed to the development of the area, thereby establishing a foundation for a more intricate comprehension of the study emphasis.Subsequently, a comprehensive examination of the extant literature pertaining directly to the research topic is conducted with great attention to detail. This process entails a rigorous amalgamation and critical analysis of scholarly publications, research works, and academic material that have been previously published. This procedure facilitates the identification of gaps, trends, and contradictions within the existing comprehension of the subject matter.

The elucidation of the idea of massive language models, which includes sophisticated artificial intelligence models such as GPT-3, is subsequently undertaken. The models have been subjected to intensive training using vast textual datasets in order to generate language that closely resembles human expression. This stage illuminates the aspects of their architectural design, training approaches, and the potential they possess for tasks related to comprehending and generating natural language.

The approaches utilized during the research process are afterwards delineated. This include the methodologies employed for the acquisition of data, the instruments and technologies utilized, the design of experiments, and any relevant procedures employed in the study. This section presents a comprehensive guide for the replication and validation of the research findings. Subsequently, the collected empirical findings obtained during the research activity are provided. This entails employing a range of analytical instruments, such as data analysis techniques, graphical representations, charts, and tables, to effectively illustrate the results. Moreover, this stage involves an in-depth analysis and exploration of the findings, elucidating their importance within the wider scope of the study.

Ultimately, the research endeavor culminates in a thorough synthesis, encompassing the fundamental insights, discoveries, and contributions that have been generated. Moreover, this particular stage serves as a foundation for future investigations by examining probable directions for additional scholarly inquiry and advancement within the field of study.

7.3 Contributions

Most significantly, our thesis work contributes by:

(a) The first contribution is collecting a coherent dataset for literature text generation in Bangla.

(b) The second contribution is the fine-tuning of of the GPT-2 model.

(c) The third contribution implements an evaluation technique with both intrinsic and extrinsic evaluation for the generated text data.

7.4 Limitations

While our study has been thorough, it does have certain limitations, which we detail below.

(a) One of the limitation of synthetic data is to generate a relational data like preserving the male-female characteristics or city-country relations etc.

(b) Major feature extraction has showed some score difference for synthetic data features with respect to the original data..

7.5 Future Works

The results achieved by our proposed methodology are really promising. Certainly, there is many area for development and several opportunities for new experiments. We have mentioned several prospective horizons below.

7.5.1 Larger Datasets

Data patterns teach NLP models, especially neural networks. Larger datasets capture language diversity, syntax, semantics, and contexts. This improves models' language generalization. Small datasets may bias NLP models. Larger datasets expose the model to more perspectives, language styles, and cultural situations, reducing bias. Small datasets may lack unusual terms, idiomatic idioms, and domain-specific terminology. Larger datasets better capture these scenarios, improving the model's language construct knowledge and use.

7.5.2 Higher Computational Resource

High-quality NLG text requires a lot of computational resources. NLG systems must process a lot of data—input text, target language, and knowledge base. Natural language understanding and machine translation require sophisticated mathematics.NLG systems need huge text and code datasets to learn. For natural-sounding writing, algorithms must learn human language patterns. sophisticated calculations: Natural language understanding and machine translation require sophisticated calculations. Large datasets make these calculations computationally demanding. NLG systems frequently generate text in real time. Systems must quickly process input and generate text.

7.5.3 Trying to modify the Language Model

There are many different architectures for language models, each with its own strengths and weaknesses. Some architectures are better suited for NLG tasks than others. For example, transformer-based models are often used for NLG tasks because they are able to learn long-range dependencies in text. The objective function is the function that the language model is optimized to minimize. The objective function for NLG tasks is typically different from the objective function for other natural language processing tasks, such as text classification or machine translation. For example, the objective function for NLG tasks may be to maximize the fluency and informativeness of the generated text. Use a different training procedure. The training procedure for language models can also be modified to improve performance for NLG

tasks. For example, it may be helpful to use a different learning rate or to train the model for a longer period of time.

7.6 Conclusion

This study presents a novel approach for the creation of artificial data to be utilized in natural language generation (NLG) challenges. The approach employed in this study is founded upon the utilization of a generative pre-trained transformer model. This model possesses the capability to acquire knowledge on the distribution of authentic data and then produce text-based data. The methodology employed in this study was assessed across a range of Natural Language Generation (NLG) tasks, and it was demonstrated that the utilization of synthetic data resulted in enhanced performance of NLG models. Nevertheless, it is important to acknowledge the limitations of our study. One constraint of our methodology is its inability to produce relational data, such as data that keeps the male-female features or city-country linkages. Another constraint of our methodology lies in its potential inability to produce data that is as precise as authentic data. In subsequent investigations, we intend to overcome the constraints of our research by formulating an approach that can produce relational data and enhance the precision of our synthetic data. Additionally, we intend to investigate the application of our methodology in other natural language generation (NLG) activities, including machine translation and text summarization.

In general, it is our contention that our methodology exhibits promise as a viable solution for the generation of synthetic data in the context of natural language generation (NLG) problems. It is anticipated that our research endeavors will contribute to the progression of natural language generation (NLG) and facilitate the creation of NLG models that are characterized by enhanced accuracy and resilience.

REFERENCES

[1] M.-F. Wong, S. Guo, C.-N. Hang, S.-W. Ho, and C.-W. Tan, "Natural language generation and understanding of big code for ai-assisted programming: A review," *Entropy*, vol. 25, no. 6, p. 888, 2023.

[2] O. Iparraguirre-Villanueva, V. Guevara-Ponce, D. Ruiz-Alvarado, S. Beltozar-Clemente, F. Sierra-Liñan, J. Zapata-Paulini, and M. Cabanillas-Carbonell, "Text prediction recurrent neural networks using long short-term memory-dropout," *Indones. J. Electr. Eng. Comput. Sci*, vol. 29, pp. 1758–1768, 2023.

[3] M. Elzohbi and R. Zhao, "Creative data generation: A review focusing on text and poetry," *arXiv preprint arXiv:2305.08493*, 2023.

[4] P. Lencastre, M. Gjersdal, L. R. Gorjão, A. Yazidi, and P. G. Lind, "Modern ai versus century-old mathematical models: How far can we go with generative adversarial networks to reproduce stochastic processes?," *Physica D: Nonlinear Phenomena*, vol. 453, p. 133831, 2023.

[5] A. K. Pandey and S. S. Roy, "Natural language generation using sequential models: A survey," *Neural Processing Letters*, pp. 1–34, 2023.

[6] Z. Zhang, H. Zhu, W. Zhang, Z. Zhang, J. Lu, K. Xu, Y. Liu, and V. Saetang, "A review of laser-induced graphene: From experimental and theoretical fabrication processes to emerging applications," *Carbon*, p. 118356, 2023.

[7] A. S. George and A. H. George, "A review of chatgpt ai's impact on several business sectors," *Partners Universal International Innovation Journal*, vol. 1, no. 1, pp. 9–23, 2023.

[8] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *Ieee Access*, vol. 7, pp. 82512–82521, 2019.

[9] O. Siméoni, M. Budnik, Y. Avrithis, and G. Gravier, "Rethinking deep active learning: Using unlabeled data at model training," in *2020 25th International conference on pattern recognition (ICPR)*, pp. 1220–1227, IEEE, 2021.

[10] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine learning techniques for multimedia*, pp. 21–49, Springer, 2008.

[11] R. He, Z. Han, X. Lu, and Y. Yin, "Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14585–14594, 2022.

[12] F. Yang, S. Zhang, W. Li, and Q. Miao, "State-of-charge estimation of lithium-ion batteries using lstm and ukf," *Energy*, vol. 201, p. 117664, 2020.

[13] F. Kratzert, "Understanding the backward pass through batch normalization layer," *Flair of Machine Learning [online]*, 2016.

[14] A. S. Rajawat and S. Jain, "Fusion deep learning based on back propagation neural network for personalization," in *2nd International Conference on Data, Engineering and Applications (IDEA)*, pp. 1–7, IEEE, 2020.

[15] S. Bhattarai, "What is gradient descent in machine learning," *A TECH BLOG*, 2018.

[16] S. Poornima and M. Pushpalatha, "Prediction of rainfall using intensified lstm based recurrent neural network with weighted linear units," *Atmosphere*, vol. 10, no. 11, p. 668, 2019.

[17] S. Chandar, C. Sankar, E. Vorontsov, S. E. Kahou, and Y. Bengio, "Towards non-saturating recurrent units for modelling long-term dependencies," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3280–3287, 2019.

[18] S. Leijnen and F. v. Veen, "The neural network zoo," *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 47, no. 1, p. 9, 2020.

[19] R. Salakhutdinov, "Learning deep generative models," *Annual Review of Statistics and Its Application*, vol. 2, pp. 361–385, 2015.

[20] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?," *arXiv preprint arXiv:1810.09136*, 2018.

[21] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang, *et al.*, "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, 2021.

[22] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13733–13742, 2021.

[23] R. Wightman, H. Touvron, and H. Jégou, "Resnet strikes back: An improved training procedure in timm," *arXiv preprint arXiv:2110.00476*, 2021.

[24] B. Koonce and B. Koonce, "Efficientnet," *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pp. 109–123, 2021.

[25] B. D. Lund and T. Wang, "Chatting about chatgpt: how may ai and gpt impact academia and libraries?," *Library Hi Tech News*, vol. 40, no. 3, pp. 26–29, 2023.

[26] D. Tsai, W. Chang, and S. Yang, "Short answer questions generation by fine-tuning bert and gpt-2," in *Proceedings of the 29th International Conference on Computers in Education Conference, ICCE*, 2021.

[27] A. P. B. Veyseh, F. Dernoncourt, B. Min, and T. H. Nguyen, "Generating complement data for aspect term extraction with gpt-2," in *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pp. 203–213, 2022.

[28] W. Yu, C. Zhu, Z. Li, Z. Hu, Q. Wang, H. Ji, and M. Jiang, "A survey of knowledge-enhanced text generation," *ACM Computing Surveys*, vol. 54, no. 11s, pp. 1–38, 2022.

[29] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang, "Towards natural language interfaces for data visualization: A survey," *IEEE transactions on visualization and computer graphics*, 2022.

[30] I. Tenney, D. Das, and E. Pavlick, "Bert rediscovers the classical nlp pipeline," *arXiv preprint arXiv:1905.05950*, 2019.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[32] N. Köbis and L. D. Mossink, "Artificial intelligence versus maya angelou: Experimental evidence that people cannot differentiate ai-generated from human-written poetry," *Computers in human behavior*, vol. 114, p. 106553, 2021.

[33] M. Hämäläinen, K. Alnajjar, and T. Poibeau, "Modern french poetry generation with roberta and gpt-2," *arXiv preprint arXiv:2212.02911*, 2022.

[34] M. Kowsher, A. A. Sami, N. J. Prottasha, M. S. Arefin, P. K. Dhar, and T. Koshiba, "Bangla-bert: transformer-based efficient model for transfer learning and language understanding," *IEEE Access*, vol. 10, pp. 91855–91870, 2022.

[35] N. Elkaref, M. Abu-Elkheir, M. ElOraby, and M. Abdelgaber, "Generating classical arabic poetry using pre-trained models," in *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pp. 53–62, 2022.

[36] H. Bunke, S. Bengio, and A. Vinciarelli, "Offline recognition of unconstrained handwritten texts using hmms and statistical language models," *IEEE transactions on Pattern analysis and Machine intelligence*, vol. 26, no. 6, pp. 709–720, 2004.

[37] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, and L. Okruszek, "Detecting formal thought disorder by deep contextualized word representations," *Psychiatry Research*, vol. 304, p. 114135, 2021.

[38] J. Guthrie, R. Marchand, and S. Marholm, "Inference of plasma parameters from fixed-bias multi-needle langmuir probes (m-nlp)," *Measurement Science and Technology*, vol. 32, no. 9, p. 095906, 2021.

[39] T. Teräsvirta, "Specification, estimation, and evaluation of smooth transition autoregressive models," *Journal of the american Statistical association*, vol. 89, no. 425, pp. 208–218, 1994.

[40] Q. Meng, D. Catchpoole, D. Skillicom, and P. J. Kennedy, "Relational autoencoder for feature extraction," in *2017 International joint conference on neural networks (IJCNN)*, pp. 364–371, IEEE, 2017.

[41] M. LLC, "MS Windows NT kernel description," 1999.

[42] G. Y. Lee, L. Alzamil, B. Doskenov, and A. Termehchy, “A survey on data cleaning methods for improved machine learning model performance,” *arXiv preprint arXiv:2109.07127*, 2021.

[43] D. Ham, J.-G. Lee, Y. Jang, and K.-E. Kim, “End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2,” in *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 583–592, 2020.

[44] X. V. Lin, T. Mihaylov, M. Artetxe, T. Wang, S. Chen, D. Simig, M. Ott, N. Goyal, S. Bhosale, J. Du, *et al.*, “Few-shot learning with multilingual generative language models,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9019–9052, 2022.

[45] L. Michelbacher, “Multi-word tokenization for natural language processing,” 2013.

[46] J. J. Webster and C. Kit, “Tokenization as the initial phase in nlp,” in *COLING 1992 volume 4: The 14th international conference on computational linguistics*, 1992.

[47] S. Pal, M. Chang, and M. F. Iriarte, “Summary generation using natural language processing techniques and cosine similarity,” in *International Conference on Intelligent Systems Design and Applications*, pp. 508–517, Springer, 2021.

[48] P.-C. Chen, H. Tsai, S. Bhojanapalli, H. W. Chung, Y.-W. Chang, and C.-S. Ferng, “A simple and effective positional encoding for transformers,” *arXiv preprint arXiv:2104.08698*, 2021.

[49] A. Gatt, F. Portet, E. Reiter, J. Hunter, S. Mahamood, W. Moncur, and S. Sripada, “From data to text in the neonatal intensive care unit: Using nlg technology for decision support and information management,” *Ai Communications*, vol. 22, no. 3, pp. 153–186, 2009.

[50] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[51] L. Fröhling and A. Zubiaga, “Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover,” *PeerJ Computer Science*, vol. 7, p. e443, 2021.

[52] O. Shliazhko, A. Fenogenova, M. Tikhonova, V. Mikhailov, A. Kozlova, and T. Shavrina, “mgpt: Few-shot learners go multilingual,” *arXiv preprint arXiv:2204.07580*, 2022.

[53] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, *et al.*, "Chatgpt for good? on opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, p. 102274, 2023.

[54] R. Ghosh, "Bangla gpt-2," 2016.

[55] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv e-prints*, 2019.

[56] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun, "Gpt-gnn: Generative pre-training of graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1857–1867, 2020.

[57] G. Frisoni, A. Carbonaro, G. Moro, A. Zammarchi, and M. Avagnano, "Nlg-metricverse: An end-to-end library for evaluating natural language generation," in *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 3465–3479, 2022.

[58] S. Baki, R. Verma, A. Mukherjee, and O. Gnawali, "Scaling and effectiveness of email masquerade attacks: Exploiting natural language generation," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 469–482, 2017.

[59] C. A. Gao, F. M. Howard, N. S. Markov, E. C. Dyer, S. Ramesh, Y. Luo, and A. T. Pearson, "Comparing scientific abstracts generated by chatgpt to real abstracts with detectors and blinded human reviewers," *NPJ Digital Medicine*, vol. 6, no. 1, p. 75, 2023.

[60] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730–27744, 2022.