**EMPOWERING EDUCATION IN NIGERIA THROUGH GENERATIVE AI: A MOBILE-FIRST APPROACH**

**BY**

**OBIKWELU KYRIAN SOCHIMA**

**20191167313**

**SUBMITTED TO**

**THE DEPARTMENT OF ELECTRICAL AND ELECTRONIC**

**ENGINEERING**

**SCHOOL OF ELECTRICAL SYSTEMS ENGINEERING AND TECHNOLOGY**

**IN PARTIAL FULFULIMENT OF THE REQUIREMENTS FOR THE**

**AWARD OF THE BACHELOR OF ENGINEERING DEGREE (B. ENG.)**

**IN ELECTRICAL AND ELECTRONIC ENGINEERING [ELECTRONIC AND COMPUTER ENGINEERING OPTION]**

**FEDERAL UNIVERSITY OF TECHNOLOGY, OWEERI**

**OCTOBER, 2024**

**CERTIFICATION PAGE**

This is to certify that this seminar work titled, “**EMPOWERING EDUCATION IN NIGERIA THROUGH GENERATIVE AI: A MOBILE-FIRST APPROACH**” was an authentic work carried work carried out by **OBIKWELU KYRIAN SOCHIMA** in partial fulfillment of the requirements for the award of the Bachelor of Engineering (B.Eng.) in Electrical and Electronic Engineering, Federal University of Technology, Owerri.

**APPROVED BY**

**…………………………….. ……………………………**

**Engr. Dr. O. C. Nosiri Date**

**(Project Supervisor)**

**DEDICATION**

This seminar report is dedicated to God, who has provided me with the strength, wisdom, and guidance necessary to complete my task. I am grateful for God's unwavering love and support, which has sustained me throughout my journey.

Additionally, this report is dedicated to my beloved parents, Mr. and Mrs. Anayo Obikwelu, and my siblings. I am grateful for your unconditional love, encouragement, and support which has been instrumental in my growth and success up to this point. Thank you so much!

**ACKNOWLEDGEMENTS**

First and foremost, I would like to express my profound gratitude to my project supervisor, Engr. Dr. O.C. Nosiri, for his expert guidance, invaluable ideas, and constant encouragement throughout the course of this study. His support has been instrumental in shaping the direction of this project.

I would also like to extend my deepest appreciation to my Head of Department, Engr. Dr. Nkwachukwu Chukwuchekwa, and to all the lecturers who have contributed to my academic growth, particularly the professors and senior faculty members whose wisdom and knowledge have been pivotal in my education.

My heartfelt thanks go to my project partners: Agu Bright, Chukwu Smart, Ndema Emmanuel, and Emenaa Alescio. Collaborating with this brilliant team has been an enriching and rewarding experience. Your dedication, insights, and contributions have made this journey a truly fruitful one.

I am immensely grateful to my parents and siblings for their unwavering support and encouragement throughout my academic journey. Your belief in me has been my greatest source of strength.

I would also like to acknowledge the open-source community for fostering my early interest in machine learning and its applications. The availability of free, open-source tools, libraries, and pre-trained machine learning models has provided invaluable resources for learning and experimentation.

Finally, to my friends, thank you for being a constant source of inspiration, support, and resourcefulness. Your presence and encouragement have been deeply appreciated.

May God bless you all!

**ABSTRACT**

This seminar report presents the architecture and applications of Generative AI, with a focus on the Transformer model. The study covers the evolution of earlier architectures such as RNNs, CNNs, and LSTMs, identifying their limitations, and introduces the Transformer model, highlighting its efficiency in addressing these issues. Key concepts like the Attention mechanism are explained in detail, along with practical applications of Transformer-based models in natural language processing and other domains. The report concludes with insights into the potential future developments in Generative AI.

**LIST OF SYMBOLS AND ABBREVIATION**

**BERT** – Bidirectional Encoder Representations from Transformers

**Bow** – Bag of Words

**CML**- Classical Machine Learning

**CNN** – Convolutional Neural Networks

**CSV** – Comma Separated Values

**DL** – Deep Learning

**FUTO** – Federal University of Technology, Owerri

**LSA** – Latent Semantic Analysis

**LSTM** – Long Short-Term Memory models

**ML** – Machine Learning

**MSE** – Mean-Squared-Error

**NLP** – Natural Language Processing

**RA** – Reference Answer

**ReLU** – Rectified Linear Unit

**RMSE** – Root Mean-Squared-Error

**RNN** – Recurrent Neural Networks

**LIST OF FIGURES AND DIAGRAMS**

[Figure 1: Diagram of Deep Neural Network 15](#_Toc177993998)

[Figure 2: Encode-Decoder Transformers Architecture 21](#_Toc177993999)

TABLE OF CONTENTS

[CERTIFICATION PAGE 1](#_Toc178041860)

[DEDICATION 2](#_Toc178041861)

[ACKNOWLEDGEMENT 3](#_Toc178041862)

[ABSTRACT 4](#_Toc178041863)

[LIST OF SYMBOLS AND ABBREVIATION 5](#_Toc178041864)

[LIST OF FIGURES AND DIAGRAMS 6](#_Toc178041865)

[TABLE OF CONTENTS 7](#_Toc178041866)

[CHAPTER ONE 8](#_Toc178041867)

[INTRODUCTION 8](#_Toc178041868)

[**1.1 Background of the Study** 8](#_Toc178041869)

[**1.2 Statement of the Problems** 9](#_Toc178041870)

[**1.3 Objectives of Study** 11](#_Toc178041871)

[**1.4 Scope of the Study** 12](#_Toc178041872)

[CHAPTER TWO: GENERATIVE AI ARCHITECURES 14](#_Toc178041873)

[**2.1.** **Introduction to Model Architectures** 14](#_Toc178041874)

[**2.2.** **Neural Networks** 14](#_Toc178041875)

[**2.3.** **Early AI Architectures** 16](#_Toc178041876)

[**2.3.1.** **Recurrent Neural Networks (RNNs)** 16](#_Toc178041877)

[**2.3.2.** **Convolutional Neural Networks (CNNs)** 17](#_Toc178041878)

[**2.3.3.** **Long Short-Term Memory Networks (LSTMs)** 18](#_Toc178041879)

[**2.4.** **The Transformer Architecture** 19](#_Toc178041880)

[**2.4.1.** **Introduction to Transformer Models** 19](#_Toc178041881)

[**2.4.2.** **Motivation for Transformers** 19](#_Toc178041882)

[**2.4.3.** **Overview of the “Attention is All You Need” Paper** 19](#_Toc178041883)

[**2.4.4.** **Detailed Architecture of Transformers** 20](#_Toc178041884)

[**2.4.5.** **Advantages of Transformers** 23](#_Toc178041885)

[**2.4.6.** **Applications of Transformers in NLP** 24](#_Toc178041886)

[REFERENCES 26](#_Toc178041887)

**CHAPTER ONE**

**INTRODUCTION**

**1.1 BACKGROUND OF THE STUDY**

Nigeria’s educational system faces significant challenges, particularly in providing quality education to its large and growing population. With an estimated population exceeding 200 million, Nigeria faces severe disparities in access to education, particularly between urban and rural areas [1]. Overcrowded classrooms, outdated learning materials, and limited access to resources hinder the educational experience for many Nigerian students.

In response to these challenges, the global education sector has increasingly turned to technology to improve learning outcomes. E-learning platforms, online courses, and adaptive learning systems have demonstrated their ability to provide personalized and flexible education [2]. The United Nations Educational, Scientific, and Cultural Organization (UNESCO) has recognized the potential of technology to enhance education, particularly in developing countries like Nigeria [3]. However, in Nigeria, integrating these technologies presents unique obstacles, including infrastructural limitations and socio-economic disparities such as limited access to stable electricity and internet connectivity.

Generative Artificial Intelligence (AI), a rapidly advancing technology, offers new opportunities for transforming education. By using AI-driven systems, personalized learning content can be created, allowing real-time adaptation of educational materials based on students’ needs and performance [4]. This technology has the potential to address some of the challenges in Nigeria’s education system, particularly in delivering updated and customized learning experiences.

At the same time, Nigeria has seen a dramatic increase in mobile phone usage and internet access with over 50% of the population using mobile phones [5]. This revolution makes a mobile-first approach to educational innovation highly relevant. This study aims to explore how generative AI, combined with mobile technology, can improve educational outcomes for Nigerian students. By focusing on the integration of AI-driven learning platforms and mobile accessibility, the research will investigate how technology can deliver high-quality, accessible education to students across the country, with the goal of bridging the gap in educational resources and access.

**1.2 STATEMENT OF THE PROBLEM**

In an era where artificial intelligence is revolutionizing global education, Nigeria finds itself at a critical crossroads. While AI-powered learning tools are increasingly commonplace in many countries [6], their integration into Nigerian classrooms remains limited and poorly adapted to local contexts. This misalignment between global technological progress and local implementation isn't just a matter of technological lag; it's a looming crisis that threatens to leave an entire generation behind.

The heart of the problem lies not in the absence of AI, but in its misalignment with Nigerian realities. Existing AI educational solutions, predominantly designed for Western educational environments, fail to address the unique challenges faced by Nigerian students and educators . These challenges include inconsistent power supply, limited internet connectivity, and the linguistic diversity of over 500 languages spoken across the nation. Furthermore, current AI tools often overlook the specific learning needs and cultural nuances of Nigerian students, potentially exacerbating rather than bridging educational gaps [7].

This research is born out of a pressing need to bridge this gap, to develop AI-powered educational tools that are not only technologically advanced but also specifically designed to be accessible, effective, and culturally relevant for Nigerian students. By addressing this critical gap, the study aims to contribute to the creation of educational solutions that can significantly improve learning outcomes and opportunities for Nigerian students in the contemporary global context.

**1.3 OBJECTIVES OF STUDY**

The main objective of the study is to develop and implement a mobile-first, AI-driven educational platform tailored to the unique challenges and opportunities present in the Nigerian educational landscape.

The specific objective include:

1. To design and develop a robust mobile application that functions efficiently on low-end smartphones, prevalent in the Nigerian market.
2. To create and curate culturally relevant educational content through the application of advanced AI models.
3. To enhance accessibility and promote inclusivity in educational technology adoption.
4. To develop and implement comprehensive metrics for assessing the platform's effectiveness in improving learning outcomes.
5. To formulate strategies for scaling the solution across different regions in Nigeria, exploring sustainable funding models through public-private partnerships, and developing frameworks for the long-term maintenance and evolution of the platform.

**1.4 SCOPE OF THE STUDY**

This research explores the potential of generative AI and mobile technology in transforming Nigeria's educational landscape. While ambitious in its goals, the study operates within defined parameters to ensure depth and feasibility. The scope encompasses several key dimensions:

1. Geographic Focus:
   1. Primary: Urban and semi-urban areas of Nigeria
   2. Target demographic: Tertiary institution students (ages 18-28)
   3. Future consideration: Expansion to rural areas and other educational levels
2. Technological Scope:
   1. Mobile platform: Android-based application for low-end smartphones
   2. AI technologies: Generative AI, natural language processing, machine learning
   3. Infrastructure: Cloud-based backend with local caching for offline functionality
3. Content Coverage:
   1. Core subjects: Mathematics, English and Sciences
   2. Supplementary materials: Lecture materials, practice tests, interactive exercises
   3. Skill development: Digital literacy, critical thinking, problem-solving modules
4. Implementation Phases:
   1. Phase 1: Research and Development.
   2. Phase 2: Pilot study in selected schools in Owerri.
   3. Phase 3: Evaluation and refinement based on user feedback
5. Partnerships and Collaborations:
   1. Government: Engagement with the Nigerian Ministry of Education
   2. NGOs: Collaborations with local and international education-focused organizations
   3. Private sector: Partnerships with Nigerian tech companies and startups
6. Research Timeline:
   1. Duration: 24 months
   2. Structure: Divided into development, implementation, and evaluation phases

While comprehensive in its approach, it's important to acknowledge the study's limitations. The focus on secondary education and urban/semi-urban areas may not fully address the unique challenges tertiary education and rural settings. However, the scope is designed to provide a robust foundation for understanding the potential of AI-driven, mobile-first educational solutions in the Nigerian context, offering valuable insights for future research and policy decisions.

**CHAPTER TWO**

**GENERATIVE AI ARCHITECTURES**

* 1. **SYSTEM MODEL ARCHITECTURES**

Model architectures in artificial intelligence (AI) refer to the structured framework of algorithms and computational components that make up an AI model. These architectures define how an AI model processes input data to produce outputs. Just as a building’s architecture dictates its appearance and functionality, an AI model’s architecture determines how it functions and performs.

The importance of model architectures cannot be overstated. They are foundational to the capabilities and efficiency of AI systems. A well-designed architecture can dramatically improve the model’s ability to learn from data, generalize from seen to unseen instances, and perform specific tasks effectively. For instance, Convolutional Neural Networks (CNNs) revolutionized image recognition tasks due to their specialized structure for handling spatial hierarchies in images [8]. Similarly, the advent of Transformer models has transformed natural language processing (NLP) by enabling efficient handling of long-range dependencies in text. [9]

In essence, the architecture dictates the “learning” process, impacting everything from computational efficiency to accuracy and applicability in real-world scenarios. Hence, understanding model architectures is crucial for developing robust, high-performing AI systems.

* 1. **NEURAL NETWORKS**

At the heart of generative AI architectures lies the concept of neural networks. These networks are the foundation upon which various model architectures are built, enabling the complex processing and generation of data. Understanding neural networks is crucial for grasping the principles behind advanced AI models, such as those used in generative tasks.

Neural networks are computational models inspired by the human brain’s structure and function. Just as the brain comprises neurons that communicate through electrical impulses, artificial neural networks consist of interconnected nodes, or artificial neurons, that process and transmit information through mathematical operations. In biological terms, a neuron receives signals through dendrites, processes these signals in the cell body, and transmits the output through an axon. Similarly, an artificial neuron receives inputs, processes them using a weighted sum, applies an activation function, and outputs the result to the next layer of neurons.

Neural networks are structured into layers, each serving a distinct purpose:

1. **Input Layer**: The input layer is where the data enters the network. Each neuron in this layer represents a feature of the input data. For instance:

* In image processing, the input layer receives pixel values of the image.
* In audio processing, it receives sampled audio data.
* In text processing, it receives tokenized text, where text is converted into numbers or tokens based on the model’s vocabulary.

1. **Hidden Layers**: These are intermediate layers that perform the core computations of the network. The hidden layers are where the network learns to extract and transform features from the input data. These layers are composed of neurons that apply weights to the inputs and pass them through activation functions. The arrangement and number of these layers vary between different model architectures, and they are crucial for the network’s performance. The hidden layers are essentially responsible for identifying patterns and relationships in the data.
2. **Output Layer**: The output layer produces the final result of the network’s computations. The structure and number of neurons in this layer depend on the specific task:

* In classification tasks, the output layer typically has a neuron for each class, outputting the probability of each class.
* In text generation, the output layer predicts the next word in a sequence, providing probabilities for each word in the vocabulary.

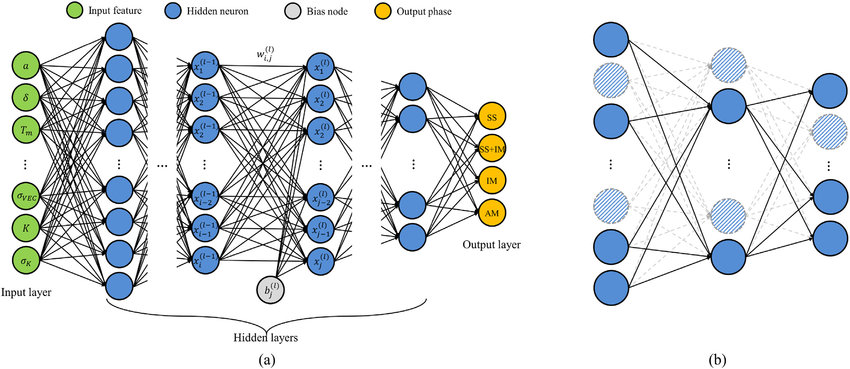


Figure 1: Diagram of Deep Neural Network

By organizing neurons into these layers, neural networks can model complex, non-linear relationships within the data, making them powerful tools for a wide range of applications.

* 1. **EARLY AI ARCHITECTURES**

The evolution of AI model architectures is a story of continuous innovation driven by the need to overcome limitations of earlier designs. Early models were simplistic, relying on basic statistical methods (linear regression, decision trees, etc.). Over time, advancements in computational power and algorithmic innovation led to the development of more sophisticated architectures, such as Recurrent Neural Networks (RNNs), Convoluted Neural Networks (CNNs), and Long Short-Term Memory Networks (LSTMs), each addressing specific challenges and expanding the potential applications of AI. Understanding this progression, pros and cons provides valuable insights into why current architectures, like Transformers, are so effective.

* + 1. **RECURRENT NEURAL NETWORKS (RNNs)**

Recurrent Neural Networks (RNNs) are designed to process sequential data by maintaining a hidden state that captures information from previous steps in the sequence. The core idea is to use this hidden state to pass information across the sequence, enabling the network to learn dependencies over time [10]

An RNN consists of an input layer, hidden layers, and an output layer. Each hidden layer receives input from both the current input and the previous hidden state, which allows the network to retain memory of past inputs. This memory aspect is what makes RNNs suitable for tasks involving sequential data.[11]

RNNs are widely used in applications that involve sequential data, such as:

1. **Sequence Prediction**: Predicting the next item in a sequence, such as stock prices or weather conditions.
2. **Language Modeling**: Predicting the next word in a sentence, useful in text generation and autocomplete features.
3. **Speech Recognition**: Converting audio signals into text by recognizing patterns in the audio sequence.
4. **Machine Translation**: Translating text from one language to another by learning from sequential dependencies in language data.

Despite their usefulness, RNNs face significant challenges:

1. **Vanishing Gradient Problem**: During training, gradients used for updating the model parameters can become very small, making it difficult for the network to learn long-term dependencies [12].
2. **Training Instability**: The dependence on sequential processing makes RNNs slower and harder to train compared to models that can process data in parallel.
3. **Short-Term Memory**: Standard RNNs struggle with retaining information over long sequences due to their limited memory capacity.
   * 1. **CONVOLUTIONAL NEURAL NETWORKS (CNNs)**

Convolutional Neural Networks (CNNs) are specialized for processing grid-like data, such as images. They use convolutional layers to automatically and adaptively learn spatial hierarchies in the data. Each convolutional layer applies a set of filters (kernels) to the input, producing feature maps that highlight various aspects of the data [13].

A typical CNN architecture includes convolutional layers, pooling layers, and fully connected layers. Convolutional layers perform the convolution operation to extract features, pooling layers reduce the dimensionality of the feature maps, and fully connected layers combine the features learned by the convolutional layers to make predictions.

CNNs excel in tasks that involve spatial data:

1. **Image Classification**: Identifying objects within an image (e.g., classifying images of cats and dogs).
2. **Object Detection**: Locating and classifying objects within an image (e.g., detecting pedestrians in an autonomous driving system).
3. **Image Segmentation**: Dividing an image into segments, such as separating foreground objects from the background.
4. **Medical Image Analysis**: Detecting anomalies in medical scans, such as tumors in MRI images.

While powerful, CNNs have limitations:

1. **Handling Sequential Data**: CNNs are not naturally suited for sequential data, such as text or time series, without significant modifications.
2. **Computational Intensity**: Training deep CNNs can be computationally expensive and requires substantial hardware resources.
3. **Interpretability**: The features learned by convolutional layers can be difficult to interpret, making it challenging to understand how the model makes decisions [14].
   * 1. **LONG SHORT-TERM MEMORY NETWORKS (LSTMs)**

Long Short-Term Memory (LSTM) networks are a special type of RNN designed to overcome the limitations of standard RNNs. LSTMs incorporate memory cells that can maintain information over long sequences, making them more effective at learning long-term dependencies [15].

An LSTM cell consists of three gates: the input gate, forget gate, and output gate. The input gate controls the extent to which new information flows into the cell, the forget gate determines what information should be discarded, and the output gate regulates the output of the cell and the information passed to the next cell. These gates allow LSTMs to retain and selectively forget information, addressing the vanishing gradient problem and improving the network’s ability to learn from long sequences [16].

LSTMs are widely used in tasks that require learning from long-term dependencies:

1. **Language Translation**: Translating text from one language to another, capturing the context over entire sentences or paragraphs.
2. **Time Series Prediction**: Forecasting future values based on past time series data, such as stock prices or weather patterns.
3. **Speech Recognition**: Converting spoken language into text by learning from sequential audio data.
4. **Text Generation**: Generating coherent text based on input sequences, such as creating captions for images.

Despite their advantages, LSTMs have some drawbacks:

1. **Computational Complexity**: LSTMs are more complex and computationally expensive to train compared to standard RNNs and CNNs.
2. **Training Time**: The additional gates and memory cells in LSTMs increase the training time and require more computational resources.
3. **Difficulty in Scaling**: Scaling LSTMs to very large datasets or extremely long sequences can be challenging.
   1. **THE TRANSFORMER ARCHITECTURE**
      1. **INTRODUCTION TO TRANSFORMER MODELS**

Transformer models, introduced by Vaswani et al. in the seminal paper “**Attention is All You Need**” [9], have revolutionized the field of Natural Language Processing (NLP) and other machine learning tasks. Prior to their development, recurrent neural networks (RNNs) and convolutional neural networks (CNNs) were the dominant architectures for handling sequential data. However, these architectures faced limitations in their ability to capture long-range dependencies, parallelize training, and handle large datasets effectively. Transformers were introduced as a novel architecture designed to address these limitations while significantly improving the performance of NLP models.

* + 1. **MOTIVATION FOR TRANSFORMERS**

The development of Transformers was motivated by the drawbacks encountered with previous sequential models, particularly RNNs and LSTMs. One of the primary challenges with these architectures was the difficulty in learning long-term dependencies due to the vanishing gradient problem. RNNs process sequences in a step-by-step manner, meaning that as the input length increases, it becomes increasingly difficult for the network to retain relevant information from earlier parts of the sequence.

Additionally, RNN-based architectures are inherently sequential, which limits their ability to parallelize computations during training. This sequential nature also makes it inefficient to process long sequences, as the model must wait for the previous step before continuing. CNNs, while effective for grid-like data such as images, also struggle with sequential data and do not naturally model relationships across different parts of a sequence.

Transformers were developed to overcome these limitations. They leverage a mechanism known as self-attention, which allows them to capture relationships between all elements of a sequence simultaneously, rather than relying on a step-by-step process. This key innovation enables Transformers to efficiently handle long-range dependencies and allows for parallelization during training, making them significantly faster and more scalable than their predecessors [17]

* + 1. **OVERVIEW OF THE “ATTENTION IS ALL YOU NEED” PAPER**

The paper “Attention is All You Need” [9] is a landmark publication in the field of machine learning and artificial intelligence. It introduced the Transformer architecture and proposed that attention mechanisms alone are sufficient for handling sequence-to-sequence tasks, eliminating the need for recurrence or convolutions. This paper is particularly notable for its application of the self-attention mechanism, which allows the model to attend to different parts of a sequence dynamically and in parallel.

The Transformer architecture outperformed RNN-based models in various NLP tasks, such as machine translation, and set new state-of-the-art results across several benchmarks. The success of this model has led to widespread adoption and the development of numerous derivatives, including Generative Pretrained Transformers (GPT), Bidirectional Encoder Representations from Transformers (BERT), and Text-to-Text Transfer Transformer (T5), which have further expanded the applications of Transformers in NLP and beyond.

* + 1. **DETAILED ARCHITECTURE OF TRANSFORMERS**

The Transformer architecture is built upon an encoder-decoder framework, but with several novel modifications that make it distinct from earlier models. The most important innovation is the introduction of the self-attention mechanism, which enables the model to process sequences in parallel and capture dependencies across distant parts of the input sequence.

The Transformer architecture consists of two main components: the encoder and the decoder. As shown in Figure 2, each of these components is composed of multiple layers of subunits, including multi-head self-attention mechanisms, feedforward neural networks, and layer normalization with residual connections.

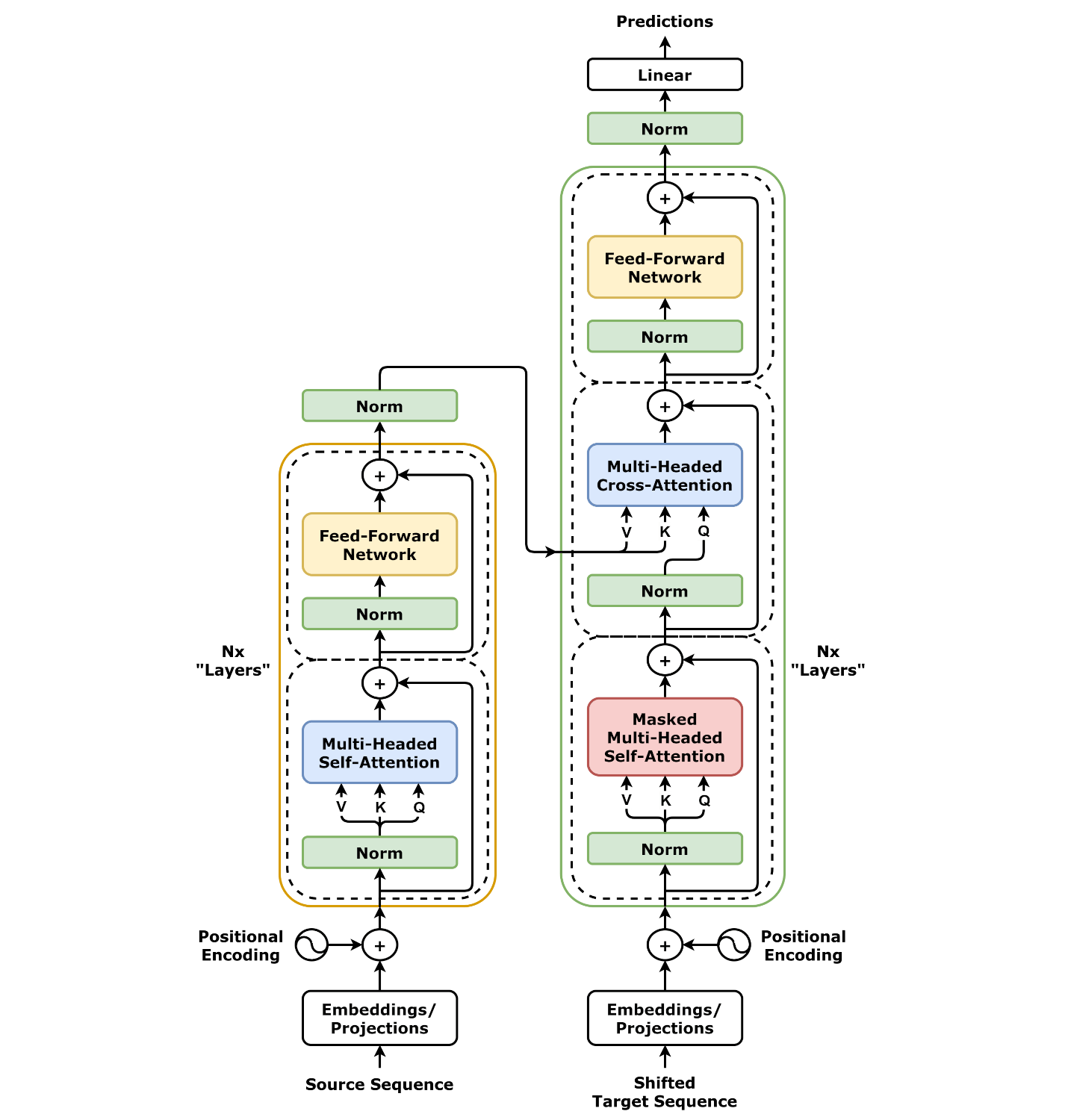


Figure 2: Encode-Decoder Transformers Architecture

1. **Encoder**

The encoder takes the input sequence and processes it through multiple layers of self-attention and feedforward networks. Each layer applies the self-attention mechanism to the input, followed by feedforward operations, layer normalization, and residual connections. The output of the encoder is a sequence of embeddings that represent the input in a transformed space, allowing the decoder to utilize this information to generate the final output.

1. **Decoder**

The decoder is responsible for generating the output sequence, such as a translation or a prediction. It also uses self-attention mechanisms, but with an additional attention layer that allows the decoder to attend to the output of the encoder. This cross-attention between the encoder and decoder enables the model to generate outputs based on the entire input sequence, even for long sequences [9].

1. **Self-Attention Mechanism**

The core innovation of the Transformer architecture is the self-attention mechanism, which allows each token in the input sequence to focus on other tokens. This is achieved by calculating a set of attention scores that measure the relevance of each token to the others in the sequence. These scores are computed using a combination of three vectors: the query, key, and value vectors. For each token in the sequence, the model computes a weighted sum of all the value vectors, where the weights are determined by the similarity between the query and key vectors [9]

Mathematically, the self-attention mechanism is computed as follows:

Where:

* represents the query matrix.
* K represents the key matrix.
* represents the value matrix.
* is the dimensionality of the key vectors.

The self-attention mechanism allows the model to capture dependencies between distant parts of the sequence without relying on the sequential processing constraints of RNNs. Moreover, Transformers use multi-head attention, where multiple attention mechanisms are applied in parallel, allowing the model to focus on different aspects of the sequence simultaneously.

1. **Positional Encoding**

Since the Transformer architecture does not inherently consider the order of the sequence, positional encoding is introduced to provide the model with information about the relative position of tokens in the sequence. This is crucial because, unlike RNNs, Transformers process sequences in parallel and therefore require an additional mechanism to understand the order of the tokens.

Positional encodings are added to the input embeddings before they are fed into the encoder and decoder layers. These encodings are generated using a set of sinusoidal functions, which allow the model to encode the position of tokens in a continuous and differentiable manner [7]. The positional encoding for each token is computed as follows:

Where:

* is the position of the token in the sequence.
* is the dimension of the embedding.
* is the total dimensionality of the embedding space.

These encodings help the model differentiate between tokens based on their positions in the sequence, allowing it to maintain an understanding of the order of inputs.

1. **Feedforward Neural Networks**

Each layer in the Transformer also includes a feedforward neural network that processes the output of the self-attention mechanism. This feedforward network consists of two fully connected layers with a ReLU activation function in between. The role of the feedforward network is to apply additional transformations to the self-attention outputs, enabling the model to capture more complex relationships in the data.

1. **Layer Normalization and Residual Connections**

To improve the training stability and performance of the model, the Transformer architecture incorporates layer normalization and residual connections. Layer normalization helps stabilize the gradients during training by ensuring that the input to each layer is normalized [18]. Residual connections allow the model to retain information from earlier layers by adding the input to the output of each sublayer, which helps mitigate the vanishing gradient problem and allows for deeper networks [19]

* + 1. **ADVANTAGES OF TRANSFORMERS**

1. **Handling Long-Range Dependencies**

One of the most significant advantages of the Transformer architecture is its ability to handle long-range dependencies effectively. Unlike RNNs and LSTMs, which struggle with retaining information over long sequences due to their sequential nature, the self-attention mechanism in Transformers allows the model to focus on all parts of the sequence simultaneously. This makes it much easier for Transformers to capture dependencies between distant tokens, improving their performance in tasks like machine translation and text generation [9]

1. **Parallelization and Training Efficiency**

The self-attention mechanism also enables parallelization during training, as the model can process all tokens in the sequence simultaneously. This is a significant improvement over RNNs, where each token must be processed sequentially. As a result, Transformers are much faster to train and can handle larger datasets more efficiently. This parallelization capability, combined with the ability to capture long-range dependencies, makes Transformers highly scalable and adaptable to various tasks.

* + 1. **APPLICATIONS OF TRANSFORMERS IN NLP**

The success of the Transformer architecture has led to its widespread adoption in numerous NLP applications, many of which have set new benchmarks for performance in their respective domains.

1. **Text Generation** - Transformers have proven to be highly effective for text generation tasks. Models like GPT use the Transformer architecture to generate coherent and contextually relevant text by predicting the next word in a sequence. These models can be fine-tuned for tasks such as story generation, dialogue systems, and content creation, making them valuable tools in the NLP space.
2. **Text-to-Text Generation -** In addition to text generation, Transformers have been applied to text-to-text generation tasks, such as machine translation and text summarization. BERT and T5 are examples of models that use the Transformer architecture to perform tasks like translating text from one language to another, summarizing long documents, or answering questions based on given passages.
3. **Question Answering -** Transformers have also demonstrated remarkable performance in question answering systems, where the model is tasked with answering questions based on a provided text. BERT, for example, was pre-trained on large amounts of text and fine-tuned for tasks like SQuAD, achieving state-of-the-art performance in question answering benchmarks.
4. **Text Classification -** Finally, Transformers are widely used in text classification tasks, where the goal is to categorize input text into predefined classes. Models like BERT and RoBERTa have achieved excellent results in sentiment analysis, spam detection, and other classification tasks by leveraging the self-attention mechanism to capture contextual relationships in the text.

**CHAPTER THREE**

**MATERIALS AND METHODS**

**3.1. MATERIALS**

1. **Data Collection**

The following data sources will be collected and used in the research:

* 1. **Primary Data**:
     + Lecture notes, materials and textbooks in the university.
     + Interviews conducted with stakeholders or experts in the field.
     + Direct observations made during field studies or experiments.
  2. **Secondary Data**:
     + Archival documents and reports related to Generative AI and Education from reputable institutions or previous research.
     + Published research papers, technical reports, and articles.
     + Statistical data obtained from relevant databases or agencies.

1. **Software Tools**

The following software tools will be used to analyze data, develop models, and perform simulations:

* 1. **Python**: For data analysis, preprocessing, and machine learning model development (e.g., libraries like NumPy, Pandas, Scikit-Learn).
  2. **Jupyter Notebooks**: For iterative analysis and documentation of research workflows.
  3. **React & React Native**: For the frontend web and mobile apps.
  4. **Microsoft Excel**: For data entry, management, and basic calculations.

1. **Hardware**

The following hardware components will be used:

* 1. **Personal Computer**: Desktop or laptop with the following minimum specifications:
     + Processor: Intel Core i7 or equivalent.
     + RAM: 16 GB (or more depending on the complexity of the simulations or data analysis).
     + Storage: 500 GB SSD for faster processing and handling of large datasets.
  2. **External storage devices**: For data backup and large file storage.
  3. **Internet Connectivity**: For access to online databases, cloud-based tools, and software resources.
  4. **METHODS**

1. **Data Preprocessing and Cleaning**
   1. Raw data from surveys, interviews, or datasets will be collected and cleaned to remove inconsistencies, missing values, and outliers.
   2. Data will be standardized or normalized where necessary to ensure uniformity across variables.
   3. Descriptive statistics will be applied to summarize key features of the data, including central tendencies (mean, median, mode) and measures of variability (standard deviation, variance).
   4. Data will be segmented based on relevant categories such as demographics, time periods, or other factors pertinent to the research.
2. **Data Analysis**
   1. Quantitative data will be analyzed using statistical methods, including correlation, regression analysis, and hypothesis testing.
   2. Qualitative data from interviews and observations will be coded and categorized based on thematic analysis.
   3. The research will include advanced machine learning or analytical models if necessary (e.g., decision trees, clustering algorithms, or deep learning models).
   4. Statistical software (e.g., SPSS, Python, R) will be employed for detailed analysis and hypothesis testing, where applicable.
3. **Model Development**
   1. Pretrained Transformers models will be finetuned for the project.
   2. Python-based simulation libraries will be used to develop models for testing scenarios and system performance.
   3. Models will be evaluated using standard performance metrics (e.g., accuracy, precision, recall) and validated through cross-validation techniques to prevent overfitting.
4. **Validation and Verification**
   1. Model validation will be conducted by testing the results against a separate validation dataset or by comparing them to real-world outcomes.
   2. Sensitivity analysis will be performed to evaluate how different variables affect the model’s performance and output.
   3. Multiple scenario analysis will be used to assess how different assumptions or conditions influence the research results.
5. **Ethical Considerations**
   1. Informed consent will be obtained from participants, and confidentiality will be maintained throughout the research process.
   2. Any sensitive data will be anonymized before analysis, and ethical guidelines for data handling will be strictly adhered to.
   3. **METHODOLOGICAL LIMITATIONS**
6. The limitations in data collection, such as sample size restrictions, potential biases in survey responses, or limited access to high-quality secondary data, will be acknowledged.
7. Challenges related to model accuracy and generalization will be discussed, particularly if the models are trained on small or unbalanced datasets.
8. The potential impact of external factors beyond the scope of this research on the outcomes will be identified.

**REFERENCES**

[1] National Population Commission, “Nigeria Population Projections,” 2021.

[2] A. Al-Zahrani, “E-learning and adaptive learning systems: trends and technologies,” Journal of e-Learning and Knowledge Society, vol. 10, no. 3, pp. 1-10, 2021.

[3] UNESCO, “Education in a Post-COVID World: Nine Ideas for Public Action,” Paris, 2021.

[4] P. Tan, “Generative AI in Education: Potential and Pitfalls,” IEEE Transactions on Learning Technologies, vol. 13, no. 2, pp. 153-162, 2022.

[5] Nigerian Communications Commission, “Telecom Subscribers Data,” 2023.

[6] M. D. Vlachopoulos and A. Makri, “The Impact of Artificial Intelligence in Education,” International Journal of Educational Research, vol. 60, no. 4, pp. 26-34, 2021.

[7] F. Adegboye, “AI-Driven Learning Systems in Nigeria: Opportunities and Barriers,” Educational Technology Review, vol. 18, no. 3, pp. 49-65, 2022.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[9] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, 2017, pp. 5998-6008.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in Neural Information Processing Systems, 2014, pp. 3104-3112.

[11] A. Graves, "Supervised sequence labelling with recurrent neural networks," Studies in Computational Intelligence, vol. 385, Springer, 2012.

[12] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in International Conference on Machine Learning, 2013, pp. 1310-1318.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, 2012, pp. 1097-1105.

[14] R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618-626.

[15] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222-2232, 2017.

[16] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," Neural Computation, vol. 12, no. 10, pp. 2451-2471, 2000.

[17] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in International Conference on Learning Representations, 2019.

[18] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer Normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.