Automated Essay Scoring System for Bangla Language

by

Tahiya Mysara Yusha
21101209
Tarannum Ahmed Nowshin
21101210
Moinuddin Zubair
21101223

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

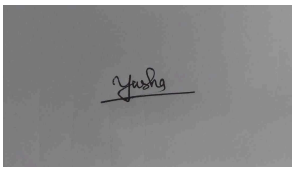
Department of Computer Science and Engineering
Brac University
June 2024

Declaration

It is hereby declared that

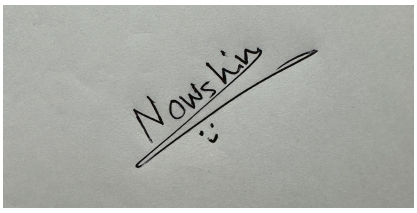
1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



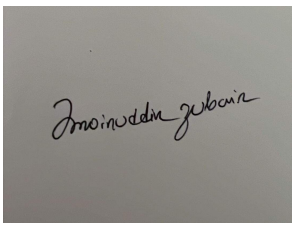
Tahiya Mysara Yusha

21101209



Tarannum Ahmed Nowshin

21101210



Moinuddin Zubair

21101223

Approval

The thesis/project titled "Automated Essay Scoring System for Bangla Language" submitted by

1. Tahiya Mysara Yusha (21101209)
2. Tarannum Ahmed Nowshin (21101210)
3. Moinuddin Zubair (21101223)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on February 03, 2025

Examining Committee:

Supervisor:

Farig Yousuf Sadeque

Dr. Farig Yousuf Sadeque
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

A plethora of scholarly articles have been thoroughly analyzed, scrutinized and delved into to decipher the growing need of Automated Essay Scoring (AES) system for Bangla and its implications on the field of education, language assessment, publishing industries, language learning platforms and other multiple distinct domains. Via a fusion of global vantage points and melding a wider international outlook with a detailed inspection on Bangladesh's local landscape, the study has identified distinct nuances and discerned correlations pertinent to the context of the nation regarding the repercussion and significance of the AES. The pressing necessity and demand for the implementation and integration of AES within the sectors of Bangladesh is palpable and exceedingly essential accentuated by the lacuna in this very critical domain. This study adeptly bridged the gap between the international trends and the intricacies of the automated essay scoring for Bangla furnishing the localized perceptions on the shifting dynamics of language assessment. Harnessing the potential of the growing Natural Language Processing (NLP) domain and making use of its powerful neural network approaches and all its alterations including the traditional transformer models like BERT and the advanced large language models (LLMs), our primary aim had been to effectively focus on these vacancies within the language assessment techniques and reshape it by modeling an automated essay scoring system for Bangla which can accurately assess and evaluate the Bangla based essays countering any biases by being completely transparent establishing a fair assessment of the dissertations and the written scripts with a commitment to elevate student learning, empowering educators and propelling advancements in educational ramifications.

Keywords: Natural Language Processing, Automated Essay System, Language assessment, Language learning platforms, Publishing industries,Transformer, Large Language Model

Acknowledgement

We would like to thank our supervisor, Dr. Farig Yousuf Sadeque, for his important direction, support, and insightful input during this study. His NLP knowledge and ongoing assistance helped shape our work. We are also grateful to Dr. Ismile Saadi for supplying us with the Bangla dataset, which helped us create our automated essay assessment model. Furthermore, we gratefully acknowledge and appreciate our mentor, Rubayat Mehjabin, a BRAC University allumni from the Department of Computer Science and Engineering, for her invaluable advice. Her earlier thesis on Automated Essay Scoring in Bangla was a valuable resource for our work, and her insightful input helped us improve our approach to many issues. Their collective help has been critical to making this study feasible, and we are grateful for their efforts.

Table of Contents

List of Figures and Tables

Nomenclature

AdamW Adaptive Moment Estimation with Weight Decay (optimization algorithm)

AES Automated Essay Scoring

AI4Bharat An open-source AI research initiative in India

AttentionMechanism A technique in neural networks that enables the model to focus on different parts of the input sequence while processing it

BERT Bidirectional Encoder Representations from Transformers

$BERT - base$ The base version of BERT model (12-layer, 768-hidden, 12-heads)

BigBird Big Bidirectional Encoder Representations from Transformers

CosSim Cosine Similarity

DL Deep Learning

Dropout A regularization technique to prevent overfitting in neural networks by randomly setting a fraction of input units to zero during training

Epoch One complete pass through the training dataset

$Fine - Tuning$ The process of training a pre-trained model on a specific task to improve its performance

GPT Generative Pretrained Transformer (a model for language generation)

HuggingFace A popular platform for NLP models and datasets, hosting implementations of models like BERT, GPT, etc.

InputEmbedding The representation of input tokens in a high-dimensional space used by transformer models

LayerNormalization A technique used to stabilize and speed up training by normalizing the activations of each layer in a model

LLM Large Language Model

LM Language Model

LSTM Long Short-Term Memory

MAE Mean Absolute Error

ML Machine Learning

$Multi-HeadAttention$ A mechanism that allows the model to focus on different parts of the input sequence simultaneously with multiple attention heads

$MuRIL$ Multilingual Representations for Indian Languages

NLP Natural Language Processing

$PretrainedModels$ Models that have been trained on large datasets and can be adapted for various tasks without starting from scratch

R^2 R-Squared (Coefficient of Determination)

$RMSE$ Root Mean Squared Error

$RMSprop$ Root Mean Squared Propagation (optimization algorithm)

$RoBERTa$ Robustly Optimized BERT Pretraining Approach

$Self-Attention$ A mechanism where each word in a sentence attends to every other word in the sentence, allowing for context-aware representations

$T5$ Text-to-Text Transfer Transformer (a transformer-based model for NLP tasks)

$Tokenizer$ A tool that splits text into tokens for model input

$Transformer$ A deep learning architecture primarily used for NLP tasks that uses attention mechanisms to process sequences

$Tuning$ The process of adjusting hyperparameters during model training to optimize performance

Chapter 1

Introduction

Technology has reshaped and advanced drastically in this era. As we had the Industrial Era in the last century which brought a remarkable change in the way people live their lives by revolutionizing the complete industrial sector. Many of the factory works which required manual labor by humans were automated which massively changed our lives and increased our efficiency and productivity. Stepping in the 20th century, a new era began which came out to be known as the Information Era. With the speedy furtherance and evolution of technology and the internet, hundreds of new inventions began to show up. One of the remarkable advancements that caught people's attention was the evolution of Artificial Intelligence. As technology began to advance and so did the research in the field of AI and Machine Learning, new breakthroughs took place which altered and ruptured every field starting from Medicine, Engineering, Financial space to Outer Space. The growing field of Automated Essay Scoring (AES) systems underwent significant research and development, similar to other domains of interest. Although AES is being studied and researched thoroughly with new inventions and technologies surrounding it, very little attention has been shed on AES systems used in Bangla even after Bangla being the 4th most verbalized language and the official language of Bangladesh. The urgent requirement for a dependable and transparent AES system for the Bangla language is the main driving force behind this research paper's objectives of revolutionizing educational evaluation and assessment in Bangla-speaking countries. In our thesis, leveraging on a substantial weighty and inimitably curated dataset, our primary objective was to present a sophisticated point of view on automated essay scoring in Bangla. One of the very central contributions of our work is the creation and amalgamation of this dataset itself which is notable and distinguished for its size and authenticity that primarily addresses the void and scarcity of the Bangla language resources. The dataset has been scrupulously and meticulously documented, annotated and graded by expert academic professionals ensuring that the assigned marks demonstrate a very high level of commitment to academic diligence and alignment with the pedagogical standards. The dataset is structured across several key columns which mainly includes the question, answer and corresponding marks forming the foundation of our analysis. Employing the advanced natural language processing (NLP) techniques (embeddings and semantic analysis) with a primary focus on these three elements, our motive had been to train the machine learning models utilizing a plethora of supervised learning techniques where the dataset's labeled instances would be used as a tool to model the relationship

between the question-answer pairs and the assigned marks, recognizing patterns and co-relations thus building a system capable of accurately predicting scores for the new sets of questions-answer pairs subsequently replicating the human evaluation process. Encouraging the creation, we tried promoting the development and implementation of Transformer models (BERT, MuRIL and AiBharat) and LLM (Llama, Falcon and Bloom). Lastly, the amalgamation of automated essay scoring systems in Bangla has been a part of our mission to amplify student learning, empower educators and advance the educational results. We put our belief in the fact that automatic essay scoring can revolutionize the field of education and steer the Bangla speaking regions toward a future where every student will have an opportunity to thrive and reach their full potential by the means of ingenuity, collaboration and a shared devotion to excellence. Ultimately, this research seeks to deliver a reliable, automated assessment of essay quality in Bangla making a substantial leap in using NLP for languages with scarce resources.

1.1 Research Problem

In this research, our motive is to address and solve the problem of the absence of an effective automated essay evaluation for the Bangla language by developing a scoring system capable of streamlining grading procedures with enhanced accurate gradings thereby filling the gap in automated assessment practices.

1.2 Research Objective

The urgent requirement for a dependable and transparent AES system for the Bangla language is the main driving force behind this research paper's objectives of revolutionizing educational evaluation and assessment in Bangla-speaking countries. Harnessing the power of technology and specifically AES systems, we look forward to mitigating the limitations and inefficiency of manual grading while unlocking new avenues for improvising the quality, efficiency and transparency of essay evaluation. With an aim to bring a change in the way Bangla essays are evaluated and graded in the educational institutions across the world backed by rigorous research, innovation and investigation our ultimate motivation is to catalyze positive change in the field of education.
The following goals serve as the cornerstone in the pursuit of developing a comprehensive, transparent and reliant Automated Essay Scoring (AES) system for Bangla:

- Developing an automated essay scoring system specifically tailored for Bangla language incorporating the unique linguistic features and cultural shade.
- Examining and investigating the current existing methodologies and approaches and evaluating their effectiveness in assessing Bangla essays.
- Evaluating the effectiveness of transformers and large language models in accurately assessing the university-level answers for Bangla essays.
- Explore the novelty and significance of our limited university-level Bangla dataset in enhancing automated grading systems.

Chapter 2

Background

2.1 Survey methodology

While selecting our research topic, we had a clear idea of what we are getting into. But to bring our vision to fruition, we had to carefully study, research and analyse the already existing work in the field of NLP that integrates the use of Transformer and Large Language Models for the evaluation of essays. Our supervisor assisted in choosing the papers which went in sync with our research. Automated Essay Evaluation in Bangla is still an unexplored field so that it has very little research around it which made us seek assistance from the rest of the papers which had works on multiple other languages aside from Bangla. Apart from it, our keen focus was fetching the papers related specifically to how NLP works and specific Transformer models such as BERT. Alongside this, we also went through some works published on the functioning and use of the Large Language Models (LLM). We regularly communicated with our supervisor regarding the information we soaked from these papers. His advice also included keeping ourselves updated with the latest trends in the NLP field. His constant encouragement allowed us to seek clarification on the unfamiliar concepts and terms that we encountered along the way.

2.2 Literature Review

The study [15] uses data augmentation techniques, particularly back translation, to increase AES performance. A straightforward technique called back translation enhances data at the document level. The authors used Google Translator to make the back translation essays. The prototype was judged on the original data and trained and verified using twice as many essay-score pairings. The authors developed a grading accommodation method for back-translation assessments in the Automated Student Assessment Prize (ASAP) dataset (https://www.kaggle.com/c/asap-aes), based on analysis of their characteristics and AES task characteristics. More than 90% of earlier studies have made use of the original ASAP dataset, which Kaggle uses to assess AES infrastructure. It has about 13,000 essays from grades 7 through 10, eight prompts, and fewer than 2,000 essays for each prompt. Studying the impact of back-translation essays on original data was the goal of the project. There were two languages used: French (single byte language) and Chinese (multi-byte language) and the Google Translate homepage version was utilized. Since back translation somewhat improves the essay's quality, the extra point was placed at 1.

Even if there was just a minor improvement, essays with high scores were given a higher score. It was believed that the essay level with the greatest frequency score would serve as the baseline. Every prompt was found to have the greatest frequency score, and essays that scored higher were awarded bonus points. To evaluate the effect of supplemented data on the model's performance, the research used two models. The second model, "Considering-Content-LSTM," calculates the KL divergence among the word arrangement of sample essays and the input essays, whereas that first model, "Manipulating-LengthGRU," divides the recurrent layer outputs by the typical essay length for every assessment. The number of misspelled terms in the ASAP dataset decreases when Chinese is used as a back translation. These mistakes may be fixed by a translation, leading to less mistakes in translated articles. The level of back-translated assessments is somewhat better than primary writings, if the translator is competent. This suggests that the translator's capacity to produce articles of a higher quality is beneficial. The study found that augmented data improved performance by 0.2% on average when compared to original data for tasks with a limited score range. writings with high scores were awarded greater scores, while writings with poor scores were kept. The impact of time augmentation was lessened as the augmented data produced the optimal model more quickly than the original data.

This research [28] provides a feature-free recurrent neural network strategy that learns the correlation between an assessment and its grading.. The system that utilizes extended short-term memory networks beats a robust baseline by 5.6% in quadratic weighted Kappa without feature engineering. For end-to-end essay grading, the system takes an essay text as input and uses recurrent neural networks to automatically learn characteristics from the data. In this paper, Tanghipour et al. (2016) primarily focus on recurrent neural networks, a highly successful machine learning model, due to their theoretical superiority and ability to learn complex patterns from data, compared to feed-forward neural networks. This AES system automatically learns characteristics and correlations between an essay's score and its recurrent neural networks. It uses non-linear neural layers to learn complicated patterns and efficiently encodes data for essay grading. The system delivers state-of-the-art performance in automated assessment scoring, surpassing a robust baseline and being among the first AES systems without hand-crafted features. The dataset used here is the one used in Kaggle's ASAP competition. Correlation or agreement measurements such as Pearson's correlation, Spearman's correlation, Kendall's Tau, and quadratic weighted Kappa (QWK) can be used to compare the output of the AES system with evaluations from human annotators (Yannakoudakis and Cummins, 2015). QWK is the formal judgment measure used in the ASAP competition; this study also employs QWK for assessment in its trials. Regardless of topic, the network learns to take essay length into account and gives short writings with less than 50 words a poor grade. This trend appears in every essay. But as essays go longer, the substance becomes more significant, and the AES system distinguishes between essays that are well-written and those that are not. The model appropriately gives well-written essays a higher score while giving other writings a lower score. This attests to the model's ability to acquire the necessary characteristics for automatic essay grading.

The goal of the study [18] was to improve artificial intelligence-based automatic scoring tools' explainability, technical complexity, and accessibility issues. Six quick engineering methodologies were used to automatically evaluate student responses using a dataset of 1,650 student responses and six essay problems. These tactics used an innovative methodology known as WRVRT to integrate zero or few-shot learning with CoT, item stem, and score rubrics. Few-shot learning performed 12.6% better than zero-shot learning, according to the study. In this paper by Lee et al. (2024), when contextual item stems and rubrics were combined with CoT prompts, scoring accuracy increased dramatically (13.44% for zero-shot and 3.7% for few-shot). Additionally, the study discovered that domain-specific reasoning improves Learning Language Models' (LLMs') performance on scoring prompts. When combined with a single-call greedy sampling or ensemble voting nucleus sampling approach, GPT-4 performed 8.64% better than GPT-3.5 in a variety of tasks. The study emphasizes how Language Processing Techniques (LLMs) may be utilized with item stems and scoring rubrics to improve automatically scored data that is understandable and comprehensible. While AI has the potential to be used for automated scoring (Zhai et al., 2020), educators are concerned with morality and transparency. According to the research, LLMs may help with this problem by making sure AI applications are understandable and comprehensible. By recognizing the sentence components in student-written replies, the GPT family, which generates natural language responses to automated scoring prompts, can determine the "black box" in scoring (Du, Liu, & Hu, 2019). Instructing GPT to utilize prompt components is another way that users may control its answer style.

The paper [16] explores how user-friendly it is to use AES methods to judge the quality of a writing. Kumar et al. (2020) analyze hand graded essays as the dataset from a 2012 contest hosted by kaggle, evaluating the suitability of automated grading tools. The study also assesses the possibilities of deep learning in AES to produce polished algorithms. The paper argues for open production data on AES research to promote comparisons and study replication. It employs deep neural networks and NLP technologies to demonstrate how rubric scores are derived linguistically, and identify key elements of an efficient rubric scoring model. The research exceeds the agreement threshold of 0.53 in prior research by finding an average degree of agreement between two human raters for each rubric and then suggests an AES system that can predict holistic essay scores and generate competitive QWKs of 0.78 which indicates the success of the study. The 1567 essay samples utilized Suite of Automatic Linguistic Analysis Tools (SALAT), GAMET, SEANCE, TAACO, TAALED, TAALES, and TAASSC, aimed to optimize the characteristics of low-level writing and and a deep learning mechanism carried out the automated process of choosing the optimal features for the AES model. Rubric scores have been modeled using regression and supervised classification approaches. The previous work on training generalizable rubric scoring models was enhanced by this one. The prior study's huge scale, short sample size, and uneven distribution of holistic and rubric scores were employed. The study's restricted feature selection procedure and use of the same training set of features made it difficult to choose the best-fitting features. Reduced rubric score scales often provide lower QWKs than bigger scales, which makes this research noteworthy. Additionally, a reduced dataset equivalent to the original training dataset was used to train, authenticate, and test the models.

The paper [3] presents an advanced version of e-rater (V.2), which is distinguished from prior automated essay grading systems by means of expert judgment-based scoring, clear and adaptable modeling techniques, a single grading model, and intuitive and relevant scoring features. Additionally, in the study, Burstein et al. (2006) offer proof of its dependability and validity. Although automated essay scoring (AES) has shown to be a competitive substitute for human scoring, its incapacity to comprehend written content continues to be a source of doubt and criticism. The defense argument contends that AES might only be effective in settings where essays are submitted in "good faith," as "bad faith" essays could be submitted by instructors or students, leading the system to give out the wrong ratings.The literature on AES has grown in response to criticisms. Three approaches have been identified by Yang et al. (2002) : the first, which compares machine-human agreement (Burstein et al., 1998; Elliot, 2001; Landauer et al., 2001), the second, which examines the relationship between test scores and other measures (Yan et al., 2001), and the third, which emphasizes understanding the scoring processes used in AES. However due to AES's data-driven methodology, which might alter often in response to various cues and situations, these studies are uncommon, which makes it challenging to talk about the significance of scores and scoring processes. AES does, however, have more promise. This study describes a novel method to AES employed in e-rater V.2, which is different from earlier versions in that attributes are combined into an essay score utilizing straightforward techniques and a limited, closely linked feature set for scoring. This method may be used to combine data from several essay prompts in a single exam to produce a single scoring model that is in line with human rubrics. The study also includes analyses based on alternate-form findings, which show a near-perfect true-score correlation and the greater reliability of e-rater ratings over human scores. To increase its validity, E-rater V.2 makes use of a small number of straightforward features. It assesses vocabulary, lexical complexity, style, development, grammar, and mechanics. The NLP foundation utilized in CriterionSM, ETS's writing education program, which recognizes the primary categories of grammar, usage, and mechanical problems, serves as the basis for this feature set. This article covers Yang's system validation techniques and validates e-rater V.2. Findings indicate that, when measuring the same concept as human scores, e-rater scores had stronger alternate-form reliability than human ratings. E-rater metrics, however, do not account for every facet of writing excellence. Detecting bad-faith essays is essential to enhancing the public's opinion of AES. Enhancing rating characteristics and offering educational criticism are continuous procedures.

With an average agreement rate of 92%, the e-rater system is an AES system created by Educational Testing Service (ETS). This study [4] looks at how well e-raters performed on essays produced by English language learners whose mother tongue is Spanish, Chinese, or Arabic in the Test of produced English (TWE). Moreover, a tiny sample of the data comes from English speakers who were born in the United States, and another comes from candidates who were born outside of the country but claimed to speak English as their first language. In this study by Burstein et al. (1999), the data were first gathered for research by Frase et al. (1997), which also included a discussion of the essay analysis. Although there were notable disparities in the ratings of English and nonnative speakers, the overall agreement rate was

comparable to operational agreement. The characteristics of the system eliminate possible problems with non-standard English syntactic or stylistic structures, and they generalize well throughout linguistic variance. E-rater analyzes 52 factors from 270 human-scored training essays to generate topic-specific models. About 92% of the time, it provides an exact matching score, which is comparable to the agreement rate between two human raters. The e-rater and human reader correlations are .73, whereas the amount is .75 between two human readers. The system's objectives are to assess essay questions for certain language groups and offer non-native English speakers individualized guidance and diagnostics. It makes use of topical analysis, discourse cue words, and syntactic characteristics. The holistic rubric criteria use an improved version of the CASS syntactic chunker (ETS) to take into account the syntactic variation of a candidate's essay replies (Abney, 1996). This study assessed the e-rater's performance on around 1100 essays from two sets of non-native speaker essay answers. The system's overall performance is strong, and the findings are extremely equivalent to those obtained when native speaker data from operational essays is scored. Syntactic, discourse, and topical analysis characteristics are among the seven or eight features used in the models developed to score TWE1 and TWE2. The models for TWE1 and TWE2 share at least four of the top five attributes found in the operational models currently in use. The majority of the characteristics chosen by the regression process were found in operational writing samples as well, indicating that the features of the system may be used for writing by non-native speakers and native speakers.

In their paper [24], Riordan et al. (2017) aimed at evaluating the individual performances of many popular neural networks with the traditional non neural methods. It breaks down how the neural networks can achieve far better outcomes than that of the robust non neural baseline models. The main emphasis has been on the neural models which can quickly bring results in accurate scoring and evaluation. The study makes use of the Unigram Embedding-based Neural Network model for short answer scoring which mainly seizes content signals from the input features. LSTM (Long Short-term Memory) is used in the case of processing the input sequences over time. The use of Convolutional Network Layer is also made to extract features from the embeddings. It also incorporates the attention mechanism module in the network architecture which involves the dot product of each LSTM hidden state with a trained vector to shed light on the most important part of the input sequence. The study also delves into the effectiveness of parameter optimization techniques for improving the accuracy of the neural models. Experimentation with these parameters can help design neural models specifically tailored for evaluating essays and short answers. Besides parameter optimization, the study also sheds light on the significance of the variability across prompts. When creating models assigned for specific tasks, a good variety of writing themes and essays should be taken into consideration.

In their paper, Shermis et al. (2013) [26] anchors on how AEE (Automated Essay Evaluation) being a multidisciplinary field embodies the research from multiple

fields that being linguistics, computer science, cognitive psychology and writing research and how focusing on how each of these fields for the development of the aimed neural network model can result in it being a more comprehensive approach. Sentiment analysis, semantic analysis, discourse coherence analysis, grammatical error detection, topic modeling and similarity detection whose core focus is comparison of the essays with a database of existing essays for the detection of plagiarism are one of the few techniques explored in this research. The prominence of construct validity in these AEE systems ensuring the alignment of the essays with the key constructs while acknowledging various adversities faced in the AEE such that of mimicking human rater behavior and besides addressing the limitations of the model is also a concern of this study.

The work of [20] sought to assess the significance of the implementation of the ethical considerations and fairness while also countering the challenges related to the automated scoring of written and verbal responses. In their work, Madnani et al. (2017) underlines the importance of adaptation of various open source tools for helping the NLP researchers into detecting, addressing and countering the biases encountered in the scoring models. The study presents us with RSMtool which is an open source Python tool which makes use of psychometric recommendations and investigates the analysis to detect the scoring biases. Warranting the program modification for fairness evaluations, collation of multiple multiple scoring model versions and appraisal of construct-irrelevant aspects are some of the course of actions by which this study promotes the importance of synergism and openness in the scoring models.

Comprehending and extracting the inferred rubrics is the main goal of this research [9].In their paper Fiacco et al.(2023) Rubrics are these methodological grading guidelines that are employed in accordance to the preset norms to gauge the caliber of the due work. The paper suggests an approach that syncs the functional components directly procured from the transformer models specifically trained for the Automated Evaluation System consisting of feature groups that is comprehensible to the humans. With an intent to refine the transparency and reliability of these AES systems, the part of study deals with extracting and aligning discrete feature groups backed by a number of extensive experiments and analysis for the validation of the method which resulted in the development of more reliable, transparent and unbiased automated scoring systems.

Presenting an approach for the AES of Bangla language essays making use of Generalized Latent Semantic Analysis (GLSA) is the main purpose of this paper [14]. In the paper, Islam suggested the AES in Bangla (ABESS) and introduced the study incorporating its assessment, inspection and system design. The objective revolves around the simple criteria of accurately assessing Bangla written essays and offering feedback to students while helping them work on their grammatical syntax, spelling and sentence coherence. This ABESS helps marking essays quite accurately em-

bodying techniques such as Generalized Latent Semantic Analysis (GLSA), Singular Value Decomposition (SVD) and other performance metrics to assess the reliability of the evaluation. Three main modules on which it works are the training essay set generation, ABESS grading module and performance evaluation module.

By utilizing the Natural Language Processing (NLP) techniques, In this paper, Singh at el.(2023) [27] aims to seek progress in the field of automated essay scoring. Using sophisticated end to end models like Transformers and LSTMs and other traditional featured based machine learning techniques, this study puts emphasis on evaluating and scoring Hindi based essays which is quite similar to that of evaluating Bangla essays.

Using cutting edge techniques and models to enhance the Bangla Automated Essay Grading (AEG), this research paper author Jiawei Liu† at el. [19] provides a way to accurately assess and mark Bangla based essays. Improving the Bangla Bert Model for the deployment of the AEG system incorporating the support of other institutions by collecting varieties of datasets is the lasting aim with a view to further extend and nurture this research.

In the paper [7], Dikli et al. (2010) introduces the idea of a two stage learning framework for Automated Essay Scoring (AES) that combines feature-engineered and end-to-end methods into the system to address the limitations of the currently existing system for essay scoring and related works. This takes semantics, coherence and prompts relevant scores into the first segment of the process using Long Short-Term Memory (LSTM) and then they are fed into a boosting tree model as the second and the last step. By the usage of this particular framework detecting the samples like essays with permuted sentences and prompt irrelevant content will be easier. The tools that are being used in this are the Long Short-Term Memory (LSTM) Neural Networks, Boosting Tree Model, eXtreme Gradient Boosting (XGBoost) and Automated Student Assessment Prize dataset (ASAP), which consists of 8 prompts. The Two-Stage Learning Framework (TSLF) is a supreme reference to develop an automated essay scoring system tailored for Bangla language essays. Detection techniques can be customized to the unique characteristics of writings in Bangla language to improve the quality of scoring correctly with the help of the two stage framework. The system that the TSLF evaluates is adaptable and efficient.

To investigate the feedback received by ESL students on their work in an automated manner, this paper [5] compares the reviews with their teachers. The objectives of the study are to assess what kind of feedback these two sources provide and to assess the efficiency with which AES systems meet the specific requirements

of ESL students. It includes about 12 ESL students who are attending an English language center in Florida, USA. In their paper ,Chen et al. (2013) , shows that the feedback provided by the teachers were shorter, focused and more positive than the ones that was provided by the AES, which were long, generic and redundant in comparison. The main data collected for this was essays from students who were from different language backgrounds and their proficiency in English language were found to be low intermediate. Among the study's resources were AES, an automated feedback system for some students, and teacher-written feedback for other students.The Internet-Based Testing-Test of English as a Foreign Language's 5-point scale writing criteria was changed to provide a comprehensive rubric that was used to evaluate the students' work.By comparing automated systems to human evaluators, the comparative study can shed light on how automated systems process feedback, which serves to understand the potential benefits and downsides of adopting automated essay scoring for Bangla.The feedback quality can be ensured, customized by the knowledge found from this paper. By establishing correlations between the results of this study and in the subject of automated essay scoring for Bangla, researchers can improve their experiments and further enhance the advancement of AES in the Bangla language context by utilizing the insights and approaches offered.

In their paper, Ramesh et al(2022), [23] proposes a listwise learning that directly incorporates the human-machine agreement into the loss function for automated essay scoring (AES). While crafting a rating model with LambdaMART, the study proposes a rank-based approach that achieves good agreement with human raters and beats state-of-the-art algorithms.The successful performance of the suggested strategy for automated essay scoring can be determined by the experimental results obtained on the ASAP dataset, which demonstrate good agreement with human raters, on pace with experienced human raters. Random Forests (RF) is an ensemble learning technique for classification and regression, while LambdaMART is a listwise learning to rank algorithm used for rating model training. Lexical, syntactical, grammar, content, fluency, and prompt-specific characteristics are among the pre-defined aspects that are utilized to pick features. Using the insights from Maximizing Human-machine Agreement, listwise learning may be included into the Automated Bangla Scoring system. Although the research primarily focuses on English essays, the techniques and ideas presented can be modified and used in the context of automated essay scoring for essays written in Bangla. The agreement between human and machine raters can also be assessed using the quadratic weighted Kappa, which can also be used for Bangla essays. Specific tools like Stanford Core NLP and Google Spelling Check API also need to be explored for processing Bangla text. Cross validation and Testing can also be done efficiently with Bangla datasets in the same manner done in the paper.

This study, Dikli et al.(2006) proposes a comprehensive evaluation of the literature on automated essay scoring systems [6]. Discussing the flaws and challenges in the body of research on automated essay grading systems is the aim of this paper.

The authors used well-known computer science repositories such as ACL, ACM, IEEE Xplore, Springer, and Science Direct and selected relevant papers based on inclusion and exclusion criteria to make a systematic review. Based on bias, external validity, and internal validity, each paper was evaluated. To find relevant research articles, the study made use of a number of digital databases and repositories, such as ACL and ACM. Essay features were extracted using NLP packages such as Word2Vec, GloVe, and NLTK.The article addresses numerous techniques, resources, and strategies applied in automated essay scoring systems which can also be used for Automated Bangla Essay Scoring. The study highlights the challenges and limitations that currently face research on automated essay grading. By having a thorough grasp of these issues, specific issues with Bangla essay scoring, such as linguistic nuances, language difficulty, and the availability of pertinent datasets can be handled .

To provide a comprehensive overview of Automated Essay Scoring (AES) systems and their application in evaluating essays, the authors , Mahjabin et al. (2024) discusses the development, implementation, and effectiveness of AES systems. The paper [21] also explores the AES technology's benefits and drawbacks in the aspect of evaluating writings of various fields, and addresses the issues of cost, time and reliability. The papers that were taken into account for this research purpose were research studies, academic papers and publications written by the experts in the field. AES systems ranging from Project Essay Grader (PEG), Intelligent Essay Assessor (IEA), E-rater, Criterion (PEG), IntelliMetric (PEG), MY Access!, and Bayesian Essay Test Scoring System (BETSY) are discussed in the piece to evaluate them in account for their features, capabilities, accuracy and reliability of scoring and applications in writing assessment. One can improve their ability to understand automated scoring and its applicability to Bangla essays by examining the techniques and resources that are used in AES systems that have been addressed in the paper. Issues such as the need for human interactions, cheating and various issues can be explored and solved and can be put into use to develop language-specific challenges in Bangla.

The current research work [2] probes the extent to which the Open-Calm LLM family grades Japanese essays using a dataset of more than 300 essays annotated by native Japanese educators covering four thematic areas, each with three different kinds of prompts. Several studies have proposed essay grading models using BERT for both English and Japanese essays; however, the performances of the models are comparable to each other. However, very large-parameter language models which have been pre-trained on huge amounts of text data are now available due to the recent success of GPT in NLP. This implies that GPTs could have worked effectively in less data-involving tasks, such as essay grading, since the proposed GPTs have a far greater parameter size than BERT. The most salient criteria for evaluation were the QWK, and the accuracy of the results were the two key factors. Going by the data in this regard, the Open-Calm Large was the most successful model that yielded a QWK score of 0.52 and an accuracy rate of 59%, while its companion, the Open-Calm Small, had lower power with a QWK score of 0.32, yielding an accuracy of 54%. Interestingly, the "Global" category had the highest accuracy percentage,

standing at 63%. Secondly, performance varied with prompts; Prompt 1 was the highest with 62%, while Prompt 3 had the lowest accuracy at 50%.

The researchers find a new approach to fine-tune the pre-trained language model with many losses of the same task to enhance the performance of the AES. In this paper [17], they propose learning text representations first using a pre-trained language model. Mean square error loss and batch-wise ListNet loss along with dynamic weights jointly constrain the scores when they are derived from representations. We evaluate our model on the Automated Student Assessment Prize dataset using QWK. The approach outperforms the state-of-the-art statistical model and state-of-the-art neural models by approximately 3%. Our model outperforms all state-of-the-art models, especially on the two narrative prompts. In the AES assignment, it is hard to build an auxiliary phrase since essay lengths are close to the BERT model's length limit. Only score labels can be accessed at present and multitask learning is challenging. They use regression loss or ranking loss, respectively, to summarize previous studies in AES. The ranking loss needs to get the exact score order, while the regression loss needs to obtain the accurate score value. Unlike the multi-task learning that uses different fully-connected networks for different tasks, several losses proposed in this paper constrain the same job to optimize the BERT model. We present R2BERT in this paper: BERT Model with Regression and Ranking. Our model learns representations of text using BERT in order to capture deep semantics. The learned representations are then mapped to scores using a fully connected neural network. Finally, the scores are jointly optimized together with dynamic combination weights, which are constrained by regression loss and batch-wise ranking loss. Our approach is tested on the open dataset of ASAP. Our model outperforms the best statistical model to date and state-of-the-art neural models at an average QWK score of about +3 percent on all eight questions, when measured using QWK.

The paper [13] evaluates the usefulness of Large Language Models (LLMs) such that of ChatGPT and Llama for the automated essay evaluation of essays. The study compares and contrasts the scores by these models against the scores by those of human raters by making use of the ASAP Dataset. The research comes to the conclusion that ChatGpt is comparatively more harsher in comparison to Llama but overall LLMs in general tend to assign very lower marks than humans. No clear correlation was found between the marking by humans and the LLM models. Another significant difference that was noted was the reluctance of the LLM models to take spelling and grammatical mistakes into account while assigning scores compared to the human counterparts. Alongside this, essay length and the substantial use of connectives had a huge impact on the overall scoring by these models. Even though LLM models like Llama showed quite promising results in scoring, they were still far away from correct assessment compared to the human raters.

This paper [12] deals with the use of the Large Language Models (LLMs) in evaluating student essays from the courses of those in workshops. The main focus during this assessment was recognition of the instilled creativity and originality. Three techniques for evaluation were taken into consideration that included pair essay comparisons, assessments based on the pre-established rubrics and also unguided

assessments where LLMs created their very own rubrics. The results showcase that LLMs on their own are very capable of generating their own rubrics but to their extreme unpredictability, their assessment variability is still an issue. Particularly in paired assessments, the quantitative analysis regarding it revealed a very moderate association between the grading by the LLMs and the human raters. Even though LLMs are more objective, they lack the experiential nuance due to which they tend to ignore each students' innate unique abilities and personal growth compared to assessments by those of the faculties.

Motivated by the increasing student-teacher ratio in Bangladesh, the paper [14] proposes a new approach to automate the evaluation of Bangla subjective answer scripts. In this context, the authors have tried to develop a reliable and efficient methodology that can assist teachers in awarding marks for descriptive answers. The system proposed in their work computes relevance, accuracy, and quality of the language used in the responses by combining keyword matching, linguistic analysis, and grammatical error detection. This system compares the responses given by students with reference answers from open and closed domains, calculates keyword frequencies, and performs some preprocessing for eliminating noise. It uses the Akkhor Bangla Spell and Grammar Checker to judge spelling and grammatical errors that go to the final score. The system comes out to be a promising basic framework when tested on 20 questions, with a relative inaccuracy of less than 10%. Shortcomings have been indicated by the authors in the Bangla resources, and ways of improvement have been suggested, such as incorporating machine learning, increased usage of parameters, and handwriting recognition for wider applications.

Chapter 3

Dataset

3.1 Description of Data

The most interesting part of this research is how the dataset was created from scratch with careful consideration to represent Automated Essay Scoring in Bangla language intricately. The initial dataset was developed by Rubayet Mahjabin (our co-supervisor) and contains 324 graded scripts from the Residential BNG103 course at BRAC University providing a rich domain for analysis. Besides that 200 essays written by a close group of students in class-8, 9 and 10 have been gathered from Dhanmondi Govt. Girls' High School on Bangladesh and Global Studies topics, along with 50 SSC first paper scripts from Monsurpur Abdul Hamid Talukder High School, broadening the educational context in a wider sense. We extend the dataset constructed by Mahjabin considerably, with 1084 more graded scripts from the course BNG103 (Residential) at BRAC University that have been collected and cleaned. While centering on university-level writing, a greater dataset of cowriting both at the school and university levels may increase confidence in our study. This research is a better contribution to the field of AES systems, by using both Mahjabin's dataset and our own in Bangla language scenario. A total of 1408 data, comprising 5 columns (FileName, Answer, Marks, Question, and Feedback), were used in our research. All of the data in the dataset come from the BNG103 course and are kept in a CSV file. Although we also have the school dataset, we have chosen to focus on the university level dataset in order to maintain our caliber. This research is particularly notable and validates its validity as a significant contribution to the field of AEG systems research because of the collection of this one-of-a-kind, unique dataset.

3.2 Preliminary Analysis

3.2.1 Pre-processing and Data Cleaning

In our research, we took into account a number of factors when pre-processing the original Bangla essays for this study because they contain extra information that isn't relevant to score prediction. The front page, logos, and names in text files were all included in the scripts from the BRAC University course BNG103. Since these elements don't meet the requirements for evaluating a Bangla essay, they are regarded as irrelevant parameters. Our co-supervisor Mahjabin purified her 324

data, eliminating any superfluous material and retaining only the marks, questions, answers, and feedback for further analysis. We did the same for the remaining 1084 data we collected. The collection, which includes scripts containing student names, is regarded as private data. Therefore, we removed all of the names and instead numbered them in the Excel sheet for data privacy reasons. Furthermore, if any of these data from the columns: Answer, Question, and Marks contained null values, we made sure to remove that entire row, as the questions and answers will be considered as the inputs and the marks will be considered as the output; thus, none can be null. For some of the answer scripts, marks were in min score 0 and max score 5, we took that and converted each and every answer script into a min score of 0 and max score of 10, so that every data can be of the same scale.

3.2.2 Tokenization

Input	Output				
Original	**Basic Tokenizer**	**Bangla BERT**	**AI Bharat**	**CSE BUET NLP**	**muRIL**
বেগম রোকেয়া তাঁর সাহিত্যকর্মের মাঝে তুলে ধরেছেন নারীর প্রতি পুরুষের অন্যায় আচরণের বিষয়টি।	['বেগম', 'রোকেয়া', 'তাঁর', 'সাহিত্যকর্মের', 'মাঝে', 'তুলে', 'ধরেছেন', 'নারীর', 'প্রতি', 'পুরুষের', 'অন্যায়', 'আচরণের', 'বিষয়টি', '।']	['বেগম', 'রে', '##োকে', '##য়া', 'তার', 'সাহিত', '##য', '##কর', '##মের', 'মাঝে', 'তলে', 'ধরেছেন', 'নারীর', 'পরত', '##ি', 'পর', '##ষের', 'অন', '##যা', '##য', 'আচরণের', 'বিষ', '##য', '##টি', '।']	['_বগ', 'ম', '_রক', 'য', '_তর', '_সহ', 'ত', 'য', 'কর', 'ম', 'র', '_ম', 'ঝ', '_তল', '_ধর', 'ছন', '_নর', 'র', '_পর', 'ত', '_পর', 'ষ', 'র', '_অন', 'য', 'য', '_আচরণ', 'র', '_ব', 'ষ', 'য', 'ট', '।']	['বেগম', '[UNK]', 'তাঁর', 'সাহিত্য', '##কর্মের', 'মাঝে', 'তুলে', 'ধরেছেন', 'নারীর', 'প্রতি', 'পুরুষের', '[UNK]', 'আচরণের', '[UNK]', '।']	['বেগম', 'রো', '##কে', '##য়া', 'তাঁর', 'সাহিত্যকর্ম', '##ের', 'মাঝে', 'তুলে', 'ধরেছেন', 'নারীর', 'প্রতি', 'পুরুষের', 'অন্যায়', 'আচরণের', 'বিষয়টি', '।']

Figure 3.1: Example of tokenizers

Basic tokenizer: Typically, a simple tokenizer divides text into words or phrases using delimiters, punctuation, and whitespace. It separates depending on punctuation or spaces because it operates at the word level and is ineffective with subwords and out-of-vocabulary (OOV) words. This tokenization is word-level and does not use sophisticated techniques to handle unfamiliar or complicated terms.

Bangla BERT tokenizer:: The model can efficiently handle both known and unknown words thanks to Sagor Sarkar's Bangla BERT's tokenizer, which is particularly made for subword tokenization. If a word is not in the model's lexicon, it is tokenized using WordPiece, which breaks words into smaller subword units. This lessens the vocabulary size and aids in handling OOV terms.

AIBharat tokenizer: The approach divides ordinary Bangla words into very small subword units. This tokenizer is meant to work with a variety of languages. However, in this case, it looks to be over-segmenting popular Bangla terms. IndicBERT may not have a large enough Bangla vocabulary, thus it breaks down frequent Bangla words into smaller subword tokens.

CSE-Buet-NLP tokenizer: Certain words or characters may be excluded from the tokenizer's vocabulary, particularly if they are uncommon or have variants (nouns), in which case such words are expressed as [UNK].

MuRIL tokenizer: For bigger words that aren't entirely in the tokenizer's vocabulary, it separates them into subword units. This method assures that no portion of the text is lost, as [UNK].The overall tokenization appears to be closer to how you would anticipate the text to be split based on Bangla grammar and meaning, with no loss of essential terms or the insertion of unfamiliar tokens.

In Mahjabin's thesis paper, they have worked with the basic tokenizer, but as we can see, it cannot handle out-of-vocabulary words. We can see the same with the CSE BUET NLP tokenizer as it replaces the OOV words with [UNK]. AIBharat does not have a large enough Bangla vocabulary. So in our case, MuRIL or Bangla BERT Model works the best for tokenization. Between these two, MuRIL works even better for tokenization for having a large Bangla vocabulary.

Large Language Model:

Llama 3.1 8B Tokenizer: LLaMA uses the SentencePiece tokenizer, which is a subword-based model that tokenizes text into smaller units without depending on whitespace as a delimiter. This is particularly helpful for languages such as Bangla, which have complex word constructions and rich morphology. The tokenizer is designed to be very efficient in tokenizing multiple languages, capturing nuances at both the word and subword levels. The LLaMA tokenizer ensures compliance with the LLaMA model's design, allowing us to evaluate its performance on Bangla essay scoring tasks using appropriately preprocessed input.

Falcon 1B and 7B Tokenizer: Falcon models rely on BPE-based byte-level tokenization, which shares a lot in common with Bloom but is oriented towards fast throughput and scalability. The tokenizer will be designed to handle open-vocabulary tasks, including those in languages containing complicated scripts, such as Bangla. It will segment the Bangla text into subwords, maintaining relevant linguistic properties in order to make sure that the input is properly digested by the model. The Falcon tokenizer supports the high vocabulary and specific script features of Bangla, which makes it suitable for testing the performance of Falcon in Bangla AES.

Bloom 560M Tokenizer : The Bloom tokenizer uses a byte-level BPE. This approach is very effective in segmenting text, especially in multilingual datasets like Bangla. It is designed to handle a wide variety of languages with complex scripts and also to break down unusual and compound words in Bangla into intelligible subword units. The Bloom tokenizer has great support for Bangla text; hence, it is suitable for testing Bloom on automated essay scoring assignments.

3.2.3 Punctuation Marks and Stopwords

We left the punctuation and stopwords in place in order to preserve the originality of the answers. We maintained the original data so that in the event that a student makes a mistake, the model can identify the error and draw conclusions from the marking of that specific essay. Since removing the stopwords could change the original data and affect the marking, we opted against doing so. For the same reason, we left the punctuations in place so that the model might pick up on any

instances where a student's grade was reduced due to a lack or excessive usage of them.

3.2.4 Data Splitting

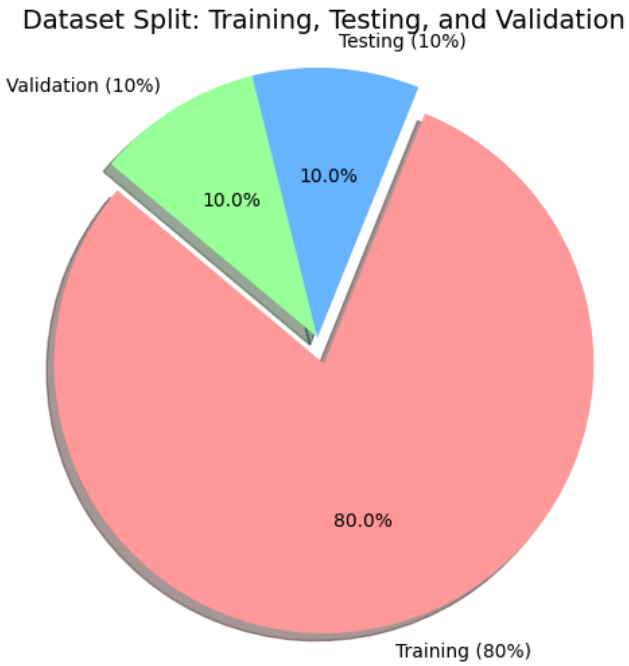


Figure 3.2: Dataset Split

The dataset that was loaded from the CSV file was segmented into three sections: training, validation, and test sets making use of the Pandas library and scikit-learn's train-test-val split function. To accomplish this division, the train test split function is used twice. Initially, it partitions the dataset into training (80%) and allocates the rest 20% for the remaining dataset fixing the random state to 42. The remaining dataset is then partitioned into testing and validation using a 50-50 split ratio, meaning that 10% is used for testing and 10% for validation.

3.2.5 Exploratory Data Analysis

This distribution graph of the Length of Answers across the dataset depicts the average length of the answers of the Bangla essays. It is evenly illustrated along the two X and Y axes. The X-axis exhibits the length of replies in terms of the word count, i.e., how many words a single answer script consists of, while the Y-axis depicts the frequency of those lengths among the answer scripts. The distribution ends up giving rise to a bell-shaped diagram, which illustrates that maximum answer length comprises between the length of 400 to 600 word counts, with the peak ranging between 550-600 words. The rise is very minimal before the 400 word count and after the 600 word count, with only a miniature number of students crossing the 1000 word feat. This diagram provides us with a very vivid picture of the word count distribution of the answer scripts across the dataset assisting with what to expect and what not to.

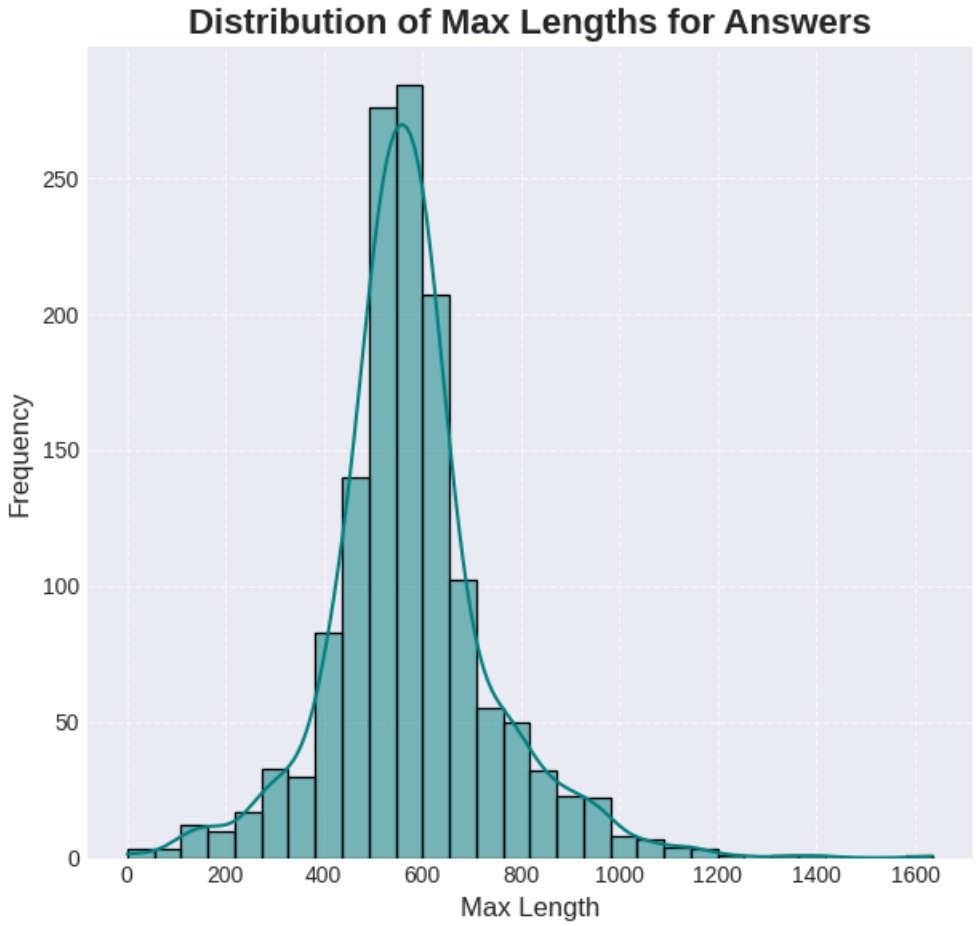


Figure 3.3: Distribution of Length of Answers across dataset

Cosine Similarity is a simple metric employed to analyze and assess the similarity and uniformity among the two non-zero vectors regardless of whatever their magnitude is. It calculates the cosine of the angle formed in between the two vectors.

The equation for Cosine Similarity is,

$$\text{Cosine Similarity (A,B)} = (\cos\theta) = \frac{A \cdot B}{|A| \cdot |B|}$$

Figure 3.4: Cosine Similarity Score

The value of cosine similarity ranges between -1 and 1. The similarity score indicates:

- **1**: The vectors are perfectly aligned, i.e., there is a resemblance between their orientation.
- **0**: The vectors are orthogonal, i.e., they have no similarity.
- **-1**: The vectors are oriented in completely opposite directions.

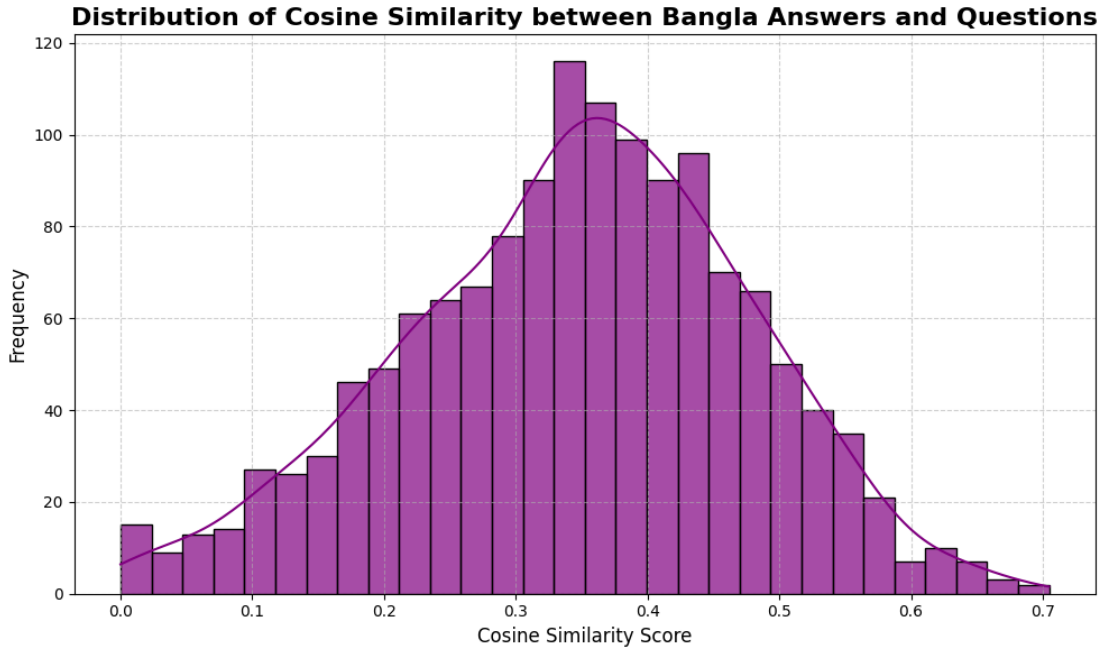


Figure 3.5: Cosine Similarity Score

In the cosine similarity graph above, the two vectors demonstrated are the questions and answers. The cosine score reveals the resemblance and similarity between the question and the answers through a calculated value. The X-axis here represents the cosine similarity score, and the Y-axis symbolizes the frequency of the scores. The lowest to highest frequency is arranged from 0 to 120. The graph shows that most of the scores range within 0.3 to 0.5, with the highest score being around 0.35 and a frequency in the vicinity of 115-120 scripts. Therefore, the answers of the students to the Bangla essay questions share an average resemblance, showing a considerable unity between the questions and the answers. A very miniature number of scripts show a score greater than 0.6, indicating that only a slight number of students actually wrote down complete relevant answers to the given question, standing around 10 students at max. Thus, this bell-shaped distribution illustrates that a good majority of the responses align with the questions, providing insights for evaluating and improving the essay scores.

This graph highlights the linkage between the essay word count and the graded marks, depicting a very positive correlation between the two. The upward sloped regression line suggests the increase in marks with the increase in the number of word count of the essays. The points that are scattered across the graph shows that the scripts with a varying number of word counts dispersed from the majority show fluctuating marks. Although, the regression line demonstrates that more word count actually assures better scores. Thus, marks ranging from 8-9 and above are promised more when the essay length is between that of 400 to 800 words. Essay length between 800 to 1000 words also reveals a very decent grade with a good majority of students attaining marks averaging 8 and above. Majority of the students with essay word length of less than 400 words and greater than that of 1000 words show average to poor performances highlighting if the word count goes down or exceeds

the 400-1000 limit, the score starts degrading. It justifies that not always more essay length guarantees better marks.

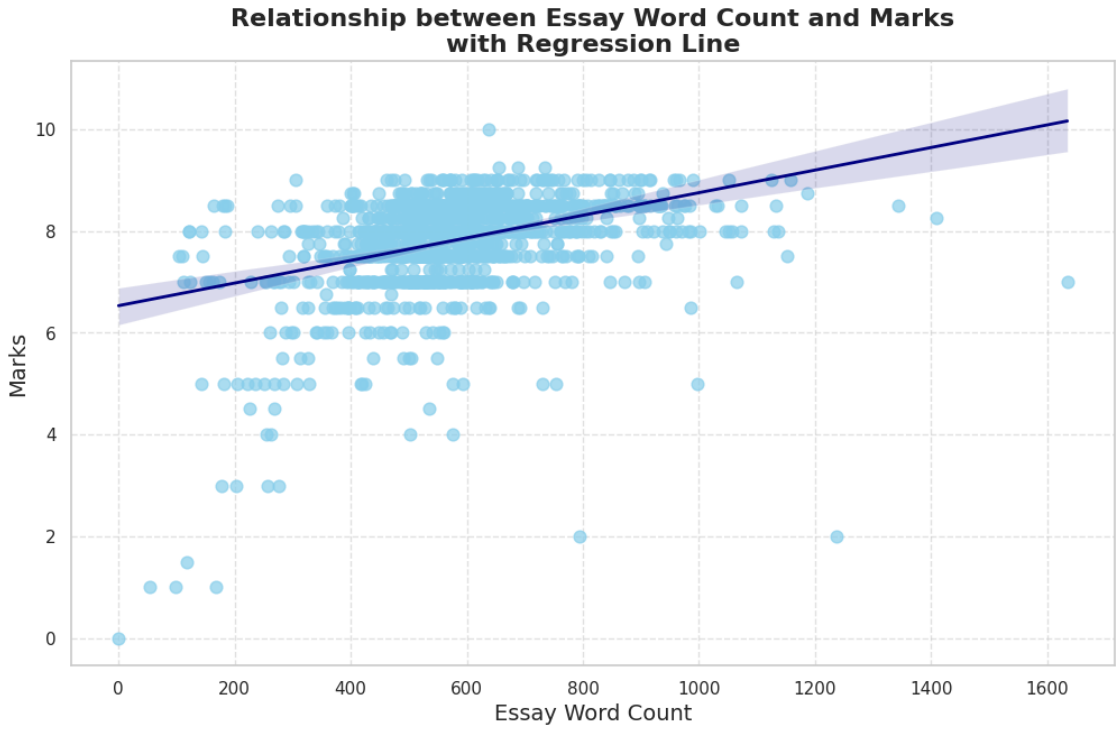


Figure 3.6: Relationship between Essay Word Count and Marks with Regression Line

The histogram bar (figure 3.7) displays the distribution of marks against frequency with the X-axis representing the marks of the students and the Y-axis illustrating the frequency of the marks. The range of score i.e. the X-axis is between 0 and 10 while the Y-axis extends from 0 to 400. The histogram illustrates that the majority of the students have a score between 7 to 9. More than 400 students scored 8 and above. 50 and above students succeeded in achieving marks ranging 9 and above. Only a handful of students scored below 6. Overall, the histogram and the overlay of Kernel Density Estimate (KDE) curve illustrates that the marks instead of being uniformly distributed are rather skewed to the right underlining the synopsis that a higher percentage of students performed brilliantly while only a small number of them underperformed.

Lexical Complexity is a tool used to identify the richness of the vocabulary and the coherence in the sentence structure. It takes various aspects of the language into account such as depth and sophistication of vocabulary and the range and frequency of word usage. It gives valuable insights into the quality of the content and helps delve deeper into identifying the caliber of the writer. In the graph (figure 3.8), the X-axis portrays the Lexical Complexity and the Y-axis represents the marks. The graph highlights that there is no such positive correlation between the complexity and sophistication of vocabulary with the marks achieved. Rather, the downhill-slope regression indicates a mild negative correlation entailing that lexical complexity has no such effects on the scores. Most essays are marked within the 7-9 range regardless of their lexical complexity, highlighting that lexical complexity has a very miniature to no effect on the marks.

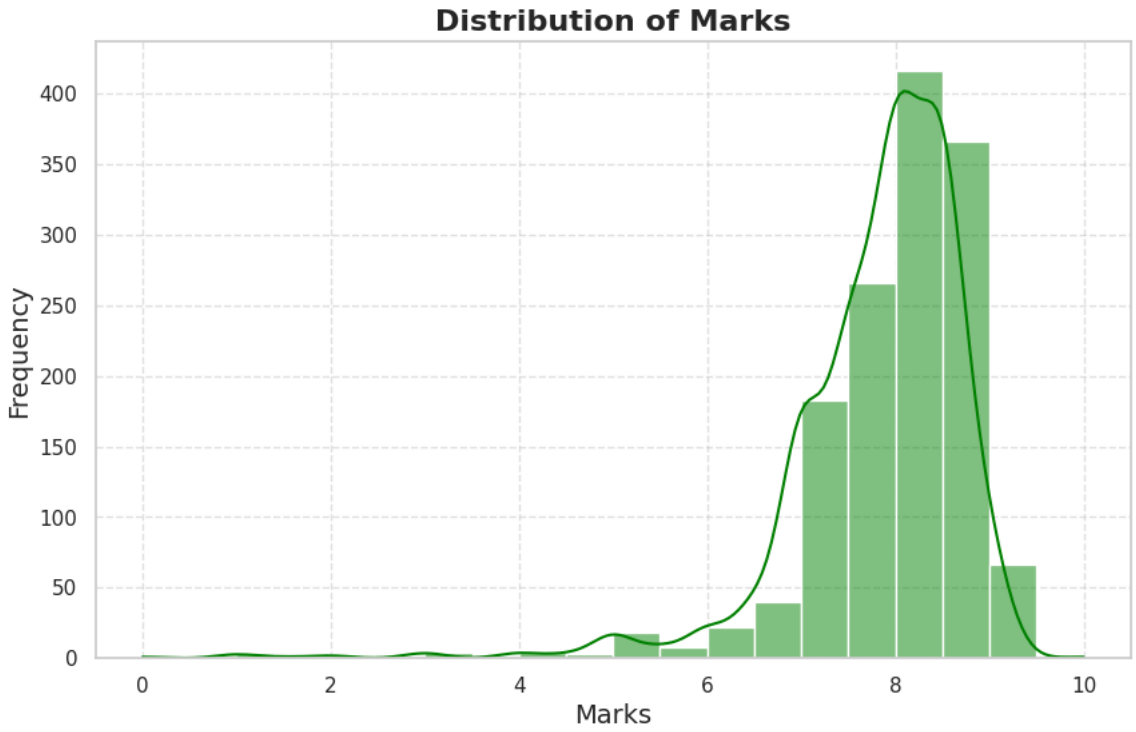


Figure 3.7: Distribution of marks

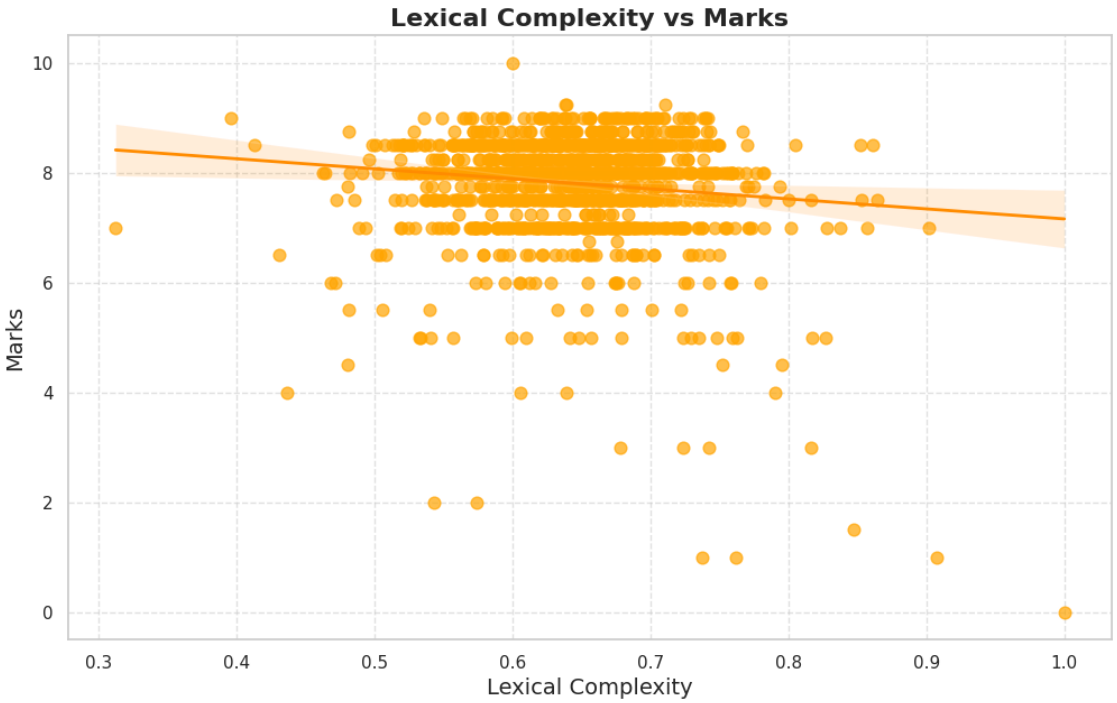


Figure 3.8: Lexical Complexity vs Marks

Chapter 4

Model

The landscape of automated essay scoring in Bangla has been significantly improved where a number of powerful and robust natural language processing (NLP) models have catered and responded to the unique distinctives and properties of the language. To serve our due purpose, a number of models stand out. After training and testing a number of them, three NLP models responded substantially well to our research objectives. The Bangla BERT model, a model specifically designed for processing Bangla based language tasks, the MuRIL model (Multilingual Representations for Indian Languages) a pre-trained multilingual transformer model developed solely for enhancing and updating the NLP capabilities across indic languages including Bangla and finally the AI Bharat model based on the Indic-Bert architecture. Apart from them, several other models were also rigorously tested and evaluated. Among them were CSE BUET Bangla, Bangla-RoBERTA, Bangla Electra and BanglaBERT-finetuned. Even though these models within themselves are sturdy and capable, they didn't yield such satisfactory results in their performance that we were expecting for our Bangla essay scoring. The grammatical analysis, syntactic accuracy and the cohesive alignment among these models did not align well with our research standards and criteria to the same extent as the chosen 3 models did. Furthermore, investigating the potential of Large Language Models (LLMs) in Bangla essay scoring is still a promising line of inquiry given the rise of models such as LLaMA, Falcon, and Bloom. We used the Llama 3.1 8B, Falcon 7B Instruct, Falcon 1B Instruct and Bloom 560M. But after training the models, we chose not to carry with the BLOOM Model due to its lack of accuracy in essay evaluation on the basis of its very unsuitable result metrics. Also a very well observed common factor in our research was how three of the chosen models are all based on BERT (Bidirectional Encoder Representations from Transformers), a revolutionary and game-changing transformer architecture that significantly elevated and refined the natural language processing tasks.

4.1 Bidirectional Encoder Representations from Transformers (BERT)

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained transformer based language model engineered by Google that has substantially boosted the natural language processing (NLP) tasks. BERT sets itself apart from

all other models due to its capability to capture context of sentences bidirectionally. That is BERT can not only process sentences in a unidirectional manner but rather it can process in both directions analyzing the context of a word is ascertained on the basis of the words that come preceding it and post the masked word in the sentence (Alammar, 2018)[2].

4.1.1 Justifications for choosing BERT over other models

BERT (Alammar, 2018) [2] is trained on an extensive dataset allowing it to be fine tuned and pre-trained for task specific activities such as essay scoring or machine translation. Apart from the ability to capture the context bidirectionally, it consists of a variety of capabilities that sets it apart from the rest of the traditional models.

Feature	*Traditional Models*	*BERT*
Contextual Understanding	Limited understanding of the context due to relying on fixed word embeddings	Deep understanding of the context due to the use of dynamic embeddings
Training Approach	Unidirectional	Bidirectional
Architecture	Simpler Architectures (e.g., LSTMs, RNNs)	Transformer Architecture
Handling of the long range dependencies	Difficulty capturing long-range dependencies	Can capture long-range dependencies
Pre-training	Usually trained on task specific data	Trained on a large corpus of dataset from varied sources
Fine-tuning	Complex	Simpler
Performance on NLP related tasks	Lower performance on complex language contemplating tasks	Achieves state-of-the-art results

Figure 4.1: BERT vs Traditional Models

4.1.2 Working Methodology of the BERT Model

1. **Start** -> The BERT process initiates.
2. **Input Text** -> Input the text data or load the CSV file.
3. **Pre-Processing** -> The input text undergoes various processes such as cleaning and normalization, including removal of special characters and tokenization preparation.
4. **Tokenization** -> The pre-processed text undergoes tokenization, i.e., it is broken down into smaller tokens. For this, BERT makes use of a special tokenization named WordPiece.
5. **Masked Language Modeling (MLM) & Next Sentence Prediction (NSP)**:

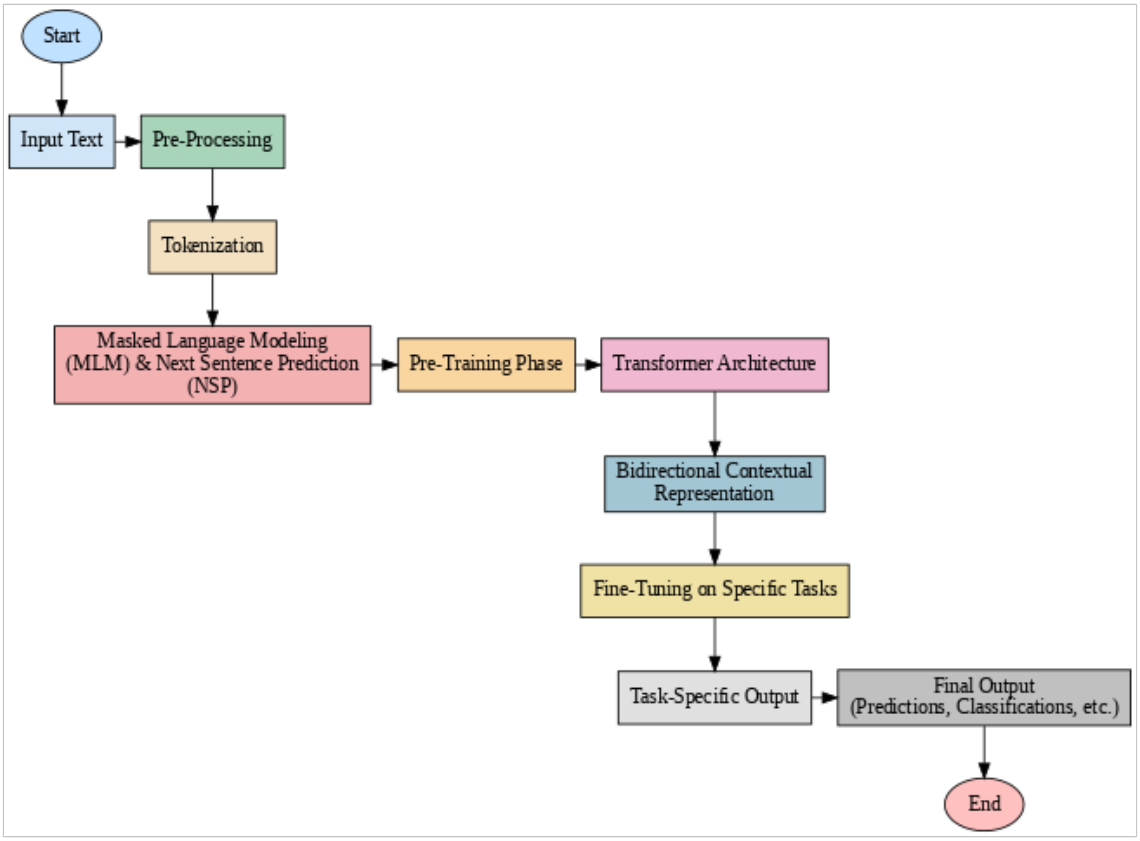


Figure 4.2: BERT

(a) **MLM** -> During the training phase, certain words in the sentence are randomly masked. BERT[19] then predicts these words based on the context of its surrounding words.

(b) **NSP** -> BERT[19] then determines whether the second sentence logically follows the first sentence, preserving sentence correlations.

6. **Pre-Training Phase** -> The BERT model then undergoes a pre-training phase where it is trained with a large corpus of data. This phase allows BERT to learn the grammar and semantics.

7. **Transformer Architecture** -> The transformer architecture is then employed by the BERT model. It consists of multiple layers of encoders, and each layer of encoder uses a self-attention mechanism to grasp the relationship between the words in the sequence.

8. **Bidirectional Contextual Representation** -> BERT then creates a bidirectional contextual representation of words, i.e., it processes the sentences from both directions, which helps it to contemplate the meaning of each word in the sentence.

9. **Fine-Tuning** -> The model is then fine-tuned on specific tasks such as named entity recognition (NER), sentiment analysis, and question answering, which refines and tailors the model's capabilities.

10. **Task Specific Output** -> BERT now produces outputs such as classifications or predictions for the specific task it was trained on.

11. **Final Output** -> BERT generates the task-specific output, for instance, the automated essay score.

12. **End** -> The BERT process terminates.

4.1.3 The Bangla BERT Model

The Sagor Sarker BERT model (Sagor Sarker, 2021)[25], also known as sagorsarker/bangla-bert-base, is a Bangla language model based on the BERT architecture. This model was created for a range of natural language processing (NLP) tasks namely text categorization, sentiment analysis and named entity recognition. This model was introduced in the year of 2021 by Sagor Sarker and his team specifically in the context of Bangla language processing in Bangladesh.

The architecture is made up of 12 layers (transformer blocks) each consisting of a hidden size encompassing 768 units and 12 attention heads. This complex structure enables the model to grasp the nuances and distinctions of the Bangla language efficiently. The model itself is pre trained on a very large corpus of data which includes books and web content. The key feature of this model is its adaptability through transfer learning which allows the users to fine-tune it using miniature dataset for specific tasks.

The Sagor Sarker BERT model is very well-suited for the "Automated Essay Scoring in Bangla" task as it is specifically designed for the Bangla language allowing it to capture the syntax, semantics and the nuances of the Bangla based texts. Using its bidirectional context processing capability, it can properly evaluate the coherence within the essays. And its transfer learning feature allows the users to fine tune the model with a specific allocated dataset just for essay scoring. Since, this model already is trained on a comprehensive corpus of data, it will only need a limited amount of data for fine-tuning thus making the research more feasible within the time constraint.

4.1.4 AI Bharat Model

The AI Bharat model (AI4Bharat,2021)[1] is based on the Indic-BERT architecture. It is specifically designed to enhance and improve the natural language processing (NLP) capabilities for many Indic languages which includes Bangla as well. This model can address a broad spectrum of tasks including question answering,sentiment analysis and text classification making it a very worthwhile model to be used for language processing tasks within the Indic languages context. This model was introduced in the year 2021 by the AI4Bharat initiative which is based in India.

It comprises 12 layers (transformer blocks) with each layer featuring a hidden size consisting of 1024 units and employs 16 attention heads. This architecture properly allows the model to grasp the diversity and nuances within the linguistic patterns

of the languages in the subcontinent. This model is trained using a substantial repository of data ensuring its relevance for the different natural language processing tasks.

A very notable feature of the AI Bharat is its ability to support transfer learning which allows the individuals to fine-tune the model using a limited set of labeled data for specific tasks. Its Indic-BERT architecture allows the model to utilize a robust framework with attention mechanisms that assists in producing rich contextual embeddings which are useful for assessing and evaluating the essays correctly.

4.1.5 The Large MuRIL Model

The MuRIL Model (Multilingual Representations for Indian Languages) (Google, 2021)[11] is a transformer driven model specifically crafted for Indian based languages such as Hindi, Tamil, Bangla and more. MuRIL is designed to tackle and address a vast array of tasks that includes sentiment analysis, named entity recognition and question-answering. This large MuRIL model was introduced in the year of 2022 by Google Research India after it introduced the base MuRIL model in 2021. The Large MuRIL model is built on the BERT Architecture which consists of 24 layers (transformer blocks) featuring a hidden size of 1024 units and 16 attention heads compared to that of the 12 layers with hidden size of 768 units and 12 attention heads of the base MuRIL model. This architecture enables it to capture the intrinsic complexities and intricacies of the Indian languages.

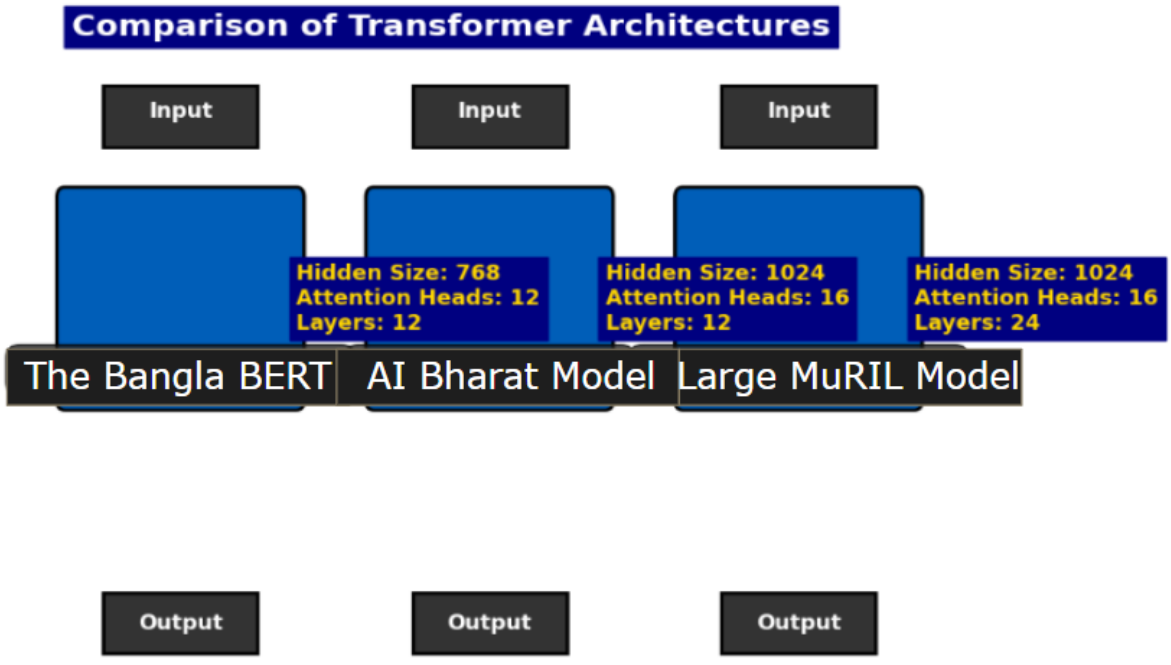


Figure 4.3: Model Architecture Comparison

Its architecture comprises 24 layers and attention mechanisms that assists in producing rich contextual embeddings enabling it to interpret the correlation between the words in a sentence which is a very essential quality for evaluating and grading the essays. Its intrinsic transfer learning capability allows the model to be

Model	*Hidden Size*	*Attention Heads*	*Layers*
Bangla *BERT* *Model*	**768**	**12**	**12**
AI Bharat Model	**1024**	**16**	**12**
Large MuRIL* *Model	**1024**	**16**	**24**

Figure 4.4: Comparison of the models

trained on a very limited amount of labeled data for meeting the demands of essay scoring thus making MuRIL an ideal contender for the essay scoring task.

4.2 Large Language Models (LLMs)

The Large Language Model, or LLM, is higher functionality artificial intelligence designed to process and produce writing that seems very much like human writing. Large language models are developed with deep learning methodologies and trained on an enormous dataset of text ingested from the internet, books, and other information repositories. The LLM can understand context, extract meaning, and perform various kinds of natural language processing tasks such as answering questions, summarizing, translation across languages, and geerating creative text. Examples of popular models that are setting a trend in changing fields like chatbots, automated writing, text analysis, and even automated scoring include GPT by OpenAI, BERT by Google, and T5. Large Language Models have huge potential in the area of automatic essay scoring, given their phenomenal strengths in natural language processing. Such models can examine text for its content, structure, coherence, grammar, and even relevance to some specific topic or question. Understanding its meaning and context, LLMs will be able to decide about the relevance of the contents with the prompt an essay gives. Besides that, LLMs consider grammar, richness of vocabulary, sentence structure, and fluency of language, along with coherence of concepts and flow to ensure logical and structural soundness. Such models are extremely scalable, with a capacity to handle the effective review of thousands of articles, hence ideal for large-scale educational contexts. LLMs could also potentially be fine-tuned on datasets corresponding to special scoring rubrics, like creativity, critical thinking, or technical precision, and hence adaptable to a wide range of educational contexts. LLMs are a relatively recent development in artificial intelligence, and their application in AES is thus still in its infancy. Unlike regular models, LLMs provide rich, context-sensitive analysis; research in this domain, however, is limited by their novelty and the focus on more general NLP tasks. Research by these usually suffers from a number of challenges: lack of big annotated essay datasets and the necessity for fine-tuning. While intriguing, additional study is required to determine their usefulness and alignment with educational requirements.

We researched Llama, Bloom, and Falcon, three advanced Large Language Models, to determine their suitability for automated essay scoring, with an emphasis on

their capacity to recognize context, assess coherence, and maintain scoring criteria.

4.2.1 Llama 3.1 8B Model

Llama 3.1 8B Instruct is an 8-billion parameter, decoder-only Transformer model. It does not have an Encoder which makes it strictly only decoder dependent. The neural network of this model contains 8 billion weights (parameters) which are adjustable in order to be used for learning the relational patterns within the data. It has in total 32 transformer layers where each layer consists of 4096 hidden units and 32 attention heads. It is developed by Meta. Its architecture makes use of the multi-head attention mechanisms for extracting the contextual information within the data. It also makes use of the feedforward neural networks for capturing the complex patterns. And in order to contemplate the word order within the sequences, it uses positional encoding.

The training process mainly comprises of two steps:

- (i) Pre-training on a large corpus of data to capture the general language understanding.
- (ii) Fine-tuning on some task specific datasets for an improved model performance.

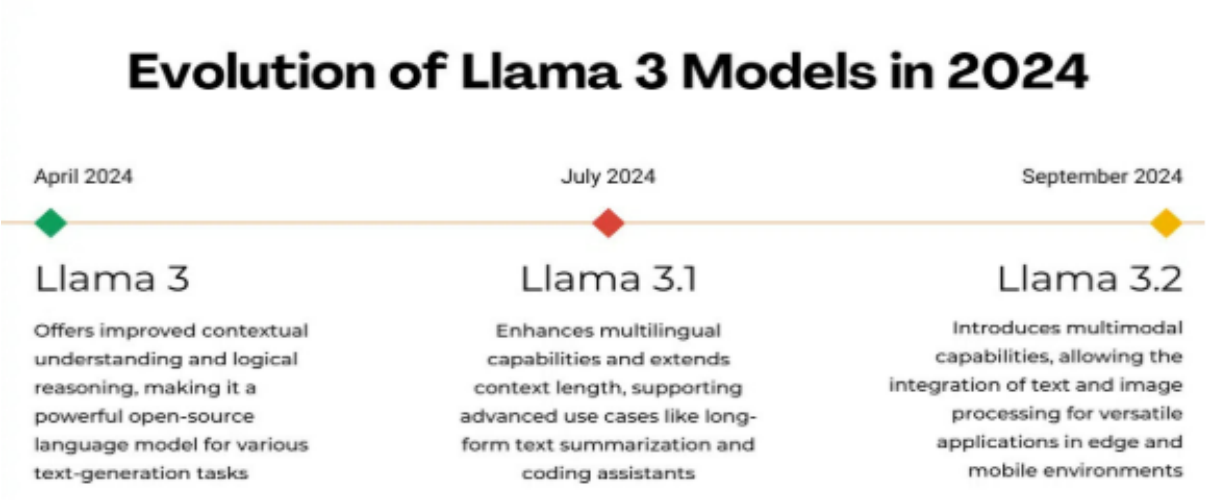


Figure 4.5: Evolution of Llama 3 Models in 2024

Alongside this for an enhanced efficiency in the computational power, Llama 3.1 incorporates the mixed precision training as well as gradient checkpointing (GeeksforGeeks, 2024)[10]. As the mode.

4.2.2 Falcon-7B-Instruct and Falcon-1B-Instruct

Falcon-7B is a casual decoder-only Transformer model that generates text after predicting the very next token in a sequence based on the previous ones. Falcon-7B-Instruct is just an alternation of the original Falcon-7B model. The only difference is that the Falcon-7B-instruct is a more fine-tuned version specifically for the instruction-following tasks. It omits the encoder part as well thus being solely

decoder dependent. Falcon 7B strictly focuses on autoregressive text generation where the next token prediction is done. It also incorporates two Attention Mechanisms namely FlashAttention and Multi-Query Attention. The architecture consists of a number of Transformer layers. Each of this layer contains its own self-attention mechanism, layer normalization and feedforward neural network. Falcon-7B-Instruct in total has 32 transformer layers where each layer consists of 4544 hidden units and 71 attention heads. The Falcon-7B also employs a rotary positional embedding (RoPE) that mainly encodes the position of the tokens to preserve the order of the words in the sequences.

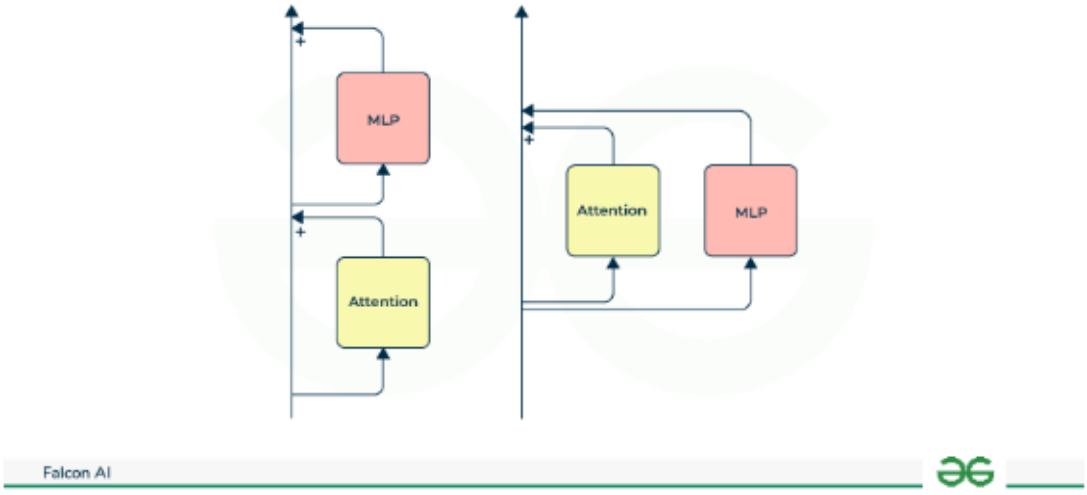


Figure 4.6: Falcon 7B and 1B instruct models

The mechanism of the model starts with the model receiving a tokenized text sequence followed by Embeddings where the tokens are converted into vector embeddings using RoPE. The next that comes is the use of the Transformer Layers that takes the input embeddings as input and extracts the semantic and contextual relationships by making use of the self-attention and feedforward networks. Finally, the model makes the prediction of the very next token in the sequence and continues this iteratively for text generation (GeeksforGeeks, 2024)[8].

Falcon-1B-Instruct follows the same mechanism but the only differences are in the number of parameters and its architecture. The total number of parameters for this model are 1 billion parameters. There are 24 transformer layers where each layer consists of 2048 hidden units and 32 attention heads. It uses 18 decoder blocks and also adopts a wider head dimensions (256).

4.3 Leveraging Quantization to Run Large Language Models (LLMs) on Limited Hardware

Large Language Models (LLMs) are typically very computationally expensive. Also, the memory requirement for it is far more than that of the traditionally encoder based Transformers. The LLama 3.1 8B has 8 billion parameters, Falcon 7B has 7 billion parameters and lastly the Falcon 1B has 1 billion parameters in total which are too huge to fit themselves within a 16GB VRAM. Quantization enables

these Large Language Models (LLMs) to fit themselves within the limited memory. Our hardware specification is 4080 Super (16 GB VRAM), that is, running these large models in full precision (FP16/FP32) would exceed our total available memory leading to out-of-memory (OOM) errors.

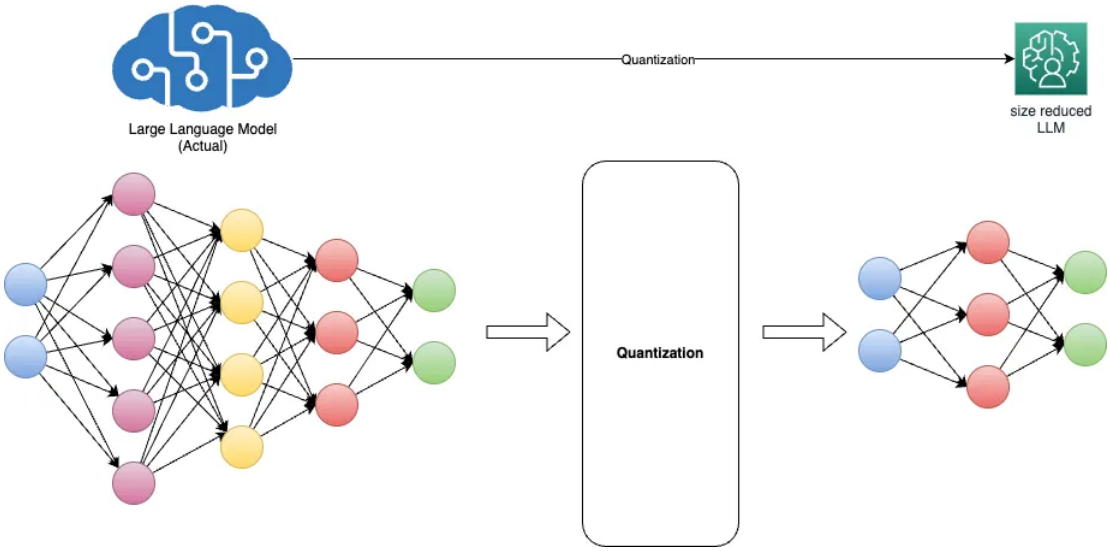


Figure 4.7: quantization

Our Rationale for Using Quantization :

Fitting the Models in GPU Memory: By making use of the 4-bit quantization (QLoRA/GGUF/GPTQ), we made the Large Language Models (LLMs) run more efficiently by reducing the overall footprint.

Improving Computational Efficiency: A higher precision model consumes enormous memory bandwidth and takes a lot of time processing the data. Using quantization allows it to turn them to lower precision models, thus enabling it to consume significantly lesser memory bandwidth and sidewise process data faster.

Enabling Fine-Tuning: It would not be feasible to run a full precision fine-tuning on a single 4080 Super. QLoRA (4-bit quantization + LoRA adapters) enabled us to fine-tune these models efficiently, which is less computationally expensive.

Preserving Model Accuracy: Even though reducing the precision can slightly impact the performance, it is very minimal. The modern quantization methods succeed in retaining most of the model's accuracy. As per our goal for automated evaluation of the essays, this trade-off was acceptable.

Our main objective of our research was comparing the performance of the traditional encoded based transformers with the Large Language Models (LLMs) for the automated evaluation of the Bangla essays. It takes huge computational power and memory to load a full LLM, train and fine-tune it. We wanted to demonstrate the effectiveness of LLMs for essay evaluations and without using quantization, it would be possible for us to load the large models in our hardware which has slight limitations. Quantization enabled us to bridge that gap and allowed us to effectively integrate the use of LLMs into our research [22].

4.4 Justification for excluding GLSA Model for AES

There has been research in AES sector of Bangla language [12], that has already been carried out where the Generalized Latent Semantic Analysis (GLSA) model has been used for assessing a number of Bangla essays. Even though it gave pretty accurate results while the assessment, we chose to not adapt the use of it for assessing the Bangla-based essays for the justified reasons described below:

Language-Specific Complications: Due to the complex syntax structure and a complicated morphology of the Bangla language, a certain number of preprocessing steps such as n-gram selection, stopwords removal, and stemming get rather difficult. As the Generalized Latent Semantic Analysis (GLSA) is developed originally for the English language, most of the time it fails to capture the complex Bangla nuances.

Limited Bangla Resources: On contrary to English, Bangla lacks most of the robust linguistic resources tailored for the ease of assessing Bangla-based essays such as advanced language models, broad-range dictionaries, and state-of-the-art corpora resources. This limitation in Bangla resources limits the GLSA model, which is strictly based in English, for smooth assessment of Bangla essays with complex syntax structure.

Contextual and Cultural Contemplation: The GLSA model lacks in effectively capturing the cultural and contextual nuances which are inherent in the Bangla language, making it difficult for the model to judge essays for more qualities outside just the grammatical accuracy.

Dimensionality Reduction Issues: The Singular Value Decomposition (SVD) that is used in the GLSA model might diminish the very subtle semantic information that is critical to the Bangla language, whereby the meaning is heavily influenced by the context and the word forms.

Our main objective of our research was comparing the performance of the traditional encoded based transformers with the Large Language Models (LLMs) for the automated evaluation of the Bangla essays. It takes huge computational power and memory to load a full LLM, train and fine-tune it. We wanted to demonstrate the effectiveness of LLMs for essay evaluations and without using quantization, it would be possible for us to load the large models in our hardware which has slight limitations. Quantization enabled us to bridge that gap and allowed us to effectively integrate the use of LLMs into our research [22]

4.5 Difference between Large Language Models and Traditional Encoder based Transformer Models

Here's a table comparing LLMs and Traditional Encoder based Transformers:

	Traditional Encoder based Transformers	LLMs (Large Language Models)
Definition	A neural network architecture for processing sequential data.	Large-scale models based on transformers, trained on extensive datasets.
Scale	Typically smaller and not pre-trained on massive datasets.	Extremely large, with billions or trillions of parameters.
Purpose	General-purpose architecture for NLP and other tasks.	Specifically designed for high-level NLP tasks like text generation and analysis.
Training	May require custom training for specific tasks.	Pre-trained on large corpora and optionally fine-tuned for specific tasks.
Examples	Transformer, Encoder-Decoder, Attention is All You Need.	GPT, BERT, LLaMA, Falcon, Bloom.
Scope	A core mechanism used in machine learning models.	A complete, pre-trained model leveraging the transformer architecture.

Figure 4.8: LLMs vs Traditional Encoder based Transformers

Chapter 5

Result analysis

5.1 Model Evaluation Metrics

To assess the model's performance, we employed three metrics: R2, RMSE and MAE.

R^2: The value of R^2 measures the proportion of variance in the actual values (score). It provides us an insight of how well our model predictions align with the actual data. The value of R^2 ranges from 0 to 1.

- When R^2 is 1, it means the model precisely predicts the real values.
- When R^2 is 0, it means the model fails to explain the variation and predict accurately.

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}} \tag{5.1}$$

Here, R^2 represents the sum of squares of residuals and TSS indicates the total sum of squares.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5.2}$$

RMSE: The value of RMSE determines how far off our predictions are from the actual values. Thus, the more the value of RMSE the bigger the error as it squares the errors first. A lower RMSE indicates better results and performance as it illustrates that our predicted values are closer to the actual values.

Range: $[0, \infty)$

- The closer the value is to 0, the less error there is.
- The farther the value is from 0, the more the error and the worse the model gets.

MAE: MAE measures how far off the predictions are actually from the actual values. It makes use of the average size of the errors to serve this purpose. It calculates the mean of the absolute difference between the predictions and the real values to find the MAE value.

- Range: $[0, \infty)$.
- The value of **MAE** is 0 for a perfect model.
- The nearer the value is to 0, the effective the model is.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5.3}$$

5.2 Model Results

5.2.1 For Traditional Encoder Models:

		Bangla Bert		*AI Bharat*		*MuRIL*	
		Smaller Dataset	Larger Dataset	Smaller Dataset	Larger Dataset	Smaller Dataset	Larger Dataset
RMSE	Validation	0.7996	0.6959	0.844	0.6879	0.7067	0.7887
	Training	0.3790	0.2015	0.695	0.4835	0.7067	0.7887
	Testing	0.8197	0.8078	0.614	0.7237	0.7586	0.8222
MAE	Validation	0.5268	0.5301	0.606	0.4704	0.6220	0.5951
	Training	0.3429	0.1789	0.545	0.3598	0.6220	0.5951
	Testing	0.5696	0.5215	0.481	0.5047	0.5787	0.6642
R^2	Validation	0.3645	0.0564	-0.80	0.2001	-0.378	0.2461
	Training	0.7641	0.9555	0.343	0.7468	-0.378	0.2461
	Testing	0.5462	0.2950	-0.45	0.2527	0.5650	0.2694

Figure 5.1: Essay Scoring Evaluation for Traditional Encoder Models

It indicates that there are differences between the performance generalizations of the Bangla Bert, AI Bharat, and MuRIL models for the training, testing, and validation datasets in smaller and bigger dataset settings. The following table summarizes their performance comparison.

The best among the various RMSEs is Bangla Bert, 0.2015 on the bigger dataset at

training, with very fine accuracy at this stage. However, testing showed big values of RMSE by Bangla Bert and AI Bharat, with a maximum of 0.8078 from Bangla Bert in the small dataset compared to that of AI Bharat's value of 0.614, while MuRIL shows generalization on this same smaller dataset of an approximate RMSE value of 0.7586. AI Bharat's performance was weaker on larger data, with a validation RMSE of 0.6879, reflecting poor predictability.

This reflects in the MAE scores: Bangla Bert has the lowest average error during training and has the lowest training MAE of 0.1789 on the bigger dataset. MuRIL, on the other hand, gives a test MAE of 0.5787 on the smaller dataset, thereby indicating a moderate amount of inaccuracy, while AI Bharat has a testing MAE of 0.481 compared to Bangla Bert's 0.5215 on the smaller dataset. The models over the larger dataset present weak generalization, with AI Bharat presenting the largest variance in this regard.

The R^2 scores show how well these models explain the variation in this data. Among them, Bangla Bert excelled in this task, which had a training R^2 of 0.9555 for the huge dataset. AI Bharat has low predicting accuracy since its test R^2 on the smaller dataset was -0.45. Low R^2 values indicate that MuRIL cannot generalize well; its performance is marginally better on the bigger dataset, achieving 0.2461 at the end of training.

The table overall reflects that Bangla Bert does quite well on training data, especially for larger sets, but it is not really doing much better on test data than AI Bharat or MuRIL. AI Bharat does poorly with larger datasets and makes huge mistakes and unstable predictions, hence is not suitable for dependable forecasts. MuRIL performs decently, but further development is required in terms of error reduction and explanation of variance to enhance predictive skills.

5.2.2 For Large Language Models (For Larger Data):

The results of the performance comparison among the LLaMA 8B, Falcon 7B, and Falcon 1B models on generalization across training, testing, and validation datasets are given in the following table.

The rest of the models were not performing as well, at 0.6880 for Falcon 1B. In second place was Falcon 7B with a validation RMSE of 0.6966, and the worst performance on the RMSE values was contributed by LLaMA 8B, with an RMSE of 0.8219. During training, the model with the poorest fittings to the training data is LLaMA 8B, with the highest RMSE of 0.9418, in contrast to Falcon 1B's 0.9204 and Falcon 7B's 0.9368. Testing RMSE also follows this pattern: the largest error is from LLaMA 8B with 1.0059, while Falcon 7B does somewhat better with 0.9391 and Falcon 1B with 0.9401.

In the patterns of MAE scores, quantified average error similarly shows the same trend. Falcon 1B has the lowest validation MAE, 0.5477, meaning it represents a better generalization on unseen data than Falcon 7B, 0.5650, and LLaMA 8B,

		Llama 8	*Falcon 7B*	*Falcon 1B*
RMSE	**Validation**	**0.8219**	**0.6966**	**0.6880**
	Training	**0.9418**	**0.9368**	**0.9204**
	Testing	**1.0059**	**0.9391**	**0.9401**
MAE	**Validation**	**0.6619**	**0.5650**	**0.5477**
	Training	**0.7236**	**0.6590**	**0.6376**
	Testing	**0.7616**	**0.6510**	**0.6391**
R^2	**Validation**	**-0.3163**	**0.0545**	**0.0779**
	Training	**0.0270**	**0.0374**	**0.0707**
	Testing	**-0.0934**	**0.0471**	**0.0452**

Figure 5.2: Essay Scoring Evaluation for Large Language Models

0.6619. Likewise, training MAE is the lowest for Falcon 1B at 0.6376, while LLaMA 8B had the largest error of 0.7236, indicating poorer training performance. Finally, the testing MAE also reflects that Falcon 1B, 0.6391, and Falcon 7B, 0.6510, outperformed LLaMA 8B with 0.7616.

The R^2 ratings, involving how well the models explain the variation, further show the struggle of LLaMA 8B. Its validation R^2 is negative at -0.3163, indicative of low predictive performance. Falcon 1B performs marginally better at 0.0779 and Falcon 7B at 0.0545. Falcon 1B does best on the training R^2 at 0.0707, followed by Falcon 7B at 0.0374 and then LLaMA 8B at 0.0270, showing that none of these models succeeded in explaining variation on train. Testing R^2 shows that LLaMA 8B has a negative score (-0.0934), while Falcon 7B (0.0471) and Falcon 1B (0.0452) are way better performers.

From the table, it could be observed that Falcon 1B has outperformed other models on all three validation, training, and testing datasets by having lower values for RMSE and MAE and a higher score for R^2. Falcon 7B stands close, although not outperforming Falcon 1B by that much. LLaMA 8B has the highest value for RMSE and MAE and negative R^2; that might suggest poor predictability and generalization.

5.3 Model predictions

5.3.1 For Traditional Encoder Models:

True vs Predicted for Traditional Encoder Models

Bangla BERT		*AI Bharat*		*MuRIL*	
True	Predicted	True	Predicted	True	Predicted
8.50	8.61	8.50	8.25	8.50	8.56
7.00	8.03	7.50	8.27	7.50	7.22
8.50	8.69	7.00	7.76	8.50	7.97
8.50	8.04	7.50	8.05	2.00	4.09
8.25	7.91	9.00	7.76	8.00	7.69
8.00	8.06	9.00	8.31	7.00	7.87
7.00	7.04	9.00	8.32	8.25	8.15
8.00	7.91	7.00	8.17	8.00	7.94
8.50	7.53	8.00	8.02	7.50	8.12

Figure 5.3: Predictions for traditional Encoder Models

5.3.2 For Large Language Models:

True vs Predicted for Large Language Models

Llama		*Falcon 7B*		*Falcon 1B*	
True	Predicted	True	Predicted	True	Predicted
8.5	7.05	8.5	7.48	8.50	7.30
7.0	6.80	7.0	7.60	7.50	7.56
7.5	7.19	7.5	7.64	7.00	7.71
7.0	7.55	7.0	7.73	5.00	7.62
7.5	7.19	7.5	7.62	7.50	7.67
5.0	6.64	7.5	7.80	7.50	8.02
7.5	7.62	7.5	7.50	6.00	7.64
7.5	7.51	3.0	7.89	8.75	7.98
8.0	7.46	8.0	7.85	7.50	7.69

Figure 5.4: Predictions for Large Language Models

5.4 Visualising Model Predictions and Error Distribution:

5.4.1 The Bangla Bert:

In this graph(figure 5.8), the training and validation losses show a drastic initial drop while converging to low values; this indicates effective learning and good generalization. Since both losses converged without divergence, there was no overfitting

Figure 5.5: Training vs validation loss for bangla bert

during training either. The model thus seems to be well optimized and capable of generalizing well to unseen data.

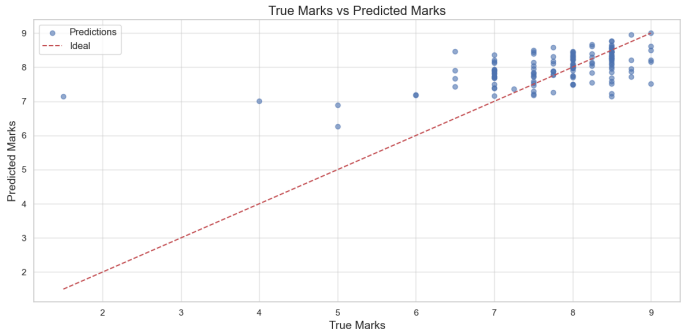


Figure 5.6: True vs predicted for Bangla BERT

This scatter plot (figure 5.9) shows the distribution of true marks versus the predicted marks; the red dotted line represents a perfect 1:1 line. As can be viewed, most points in the mark range between 6 and 8 are lying close to the ideal line, meaning in this range of marks, the true values are correctly forecasted. However, for lower true marks, the model tends to overestimate, as shown by the upward deviation of points from the ideal line for lower true marks, such as below 5. This would suggest that the model performs better for higher scores than for lower ones.

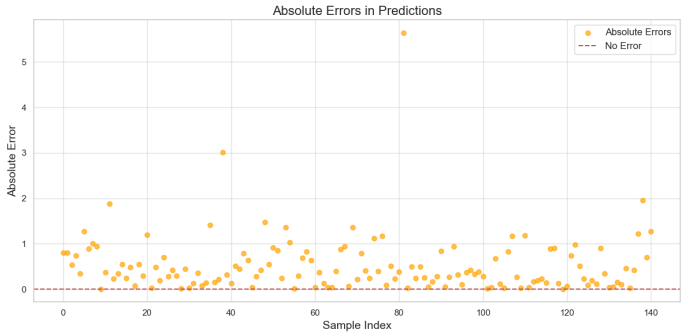


Figure 5.7: Absolute error in prediction for bangla bert

This is the absolute error for each prediction plotted by sample index. Most errors are small, indicating near perfect predictions. However, a few outliers above 3 indicate that this is where the model can be improved

5.4.2 AI BHARAT:

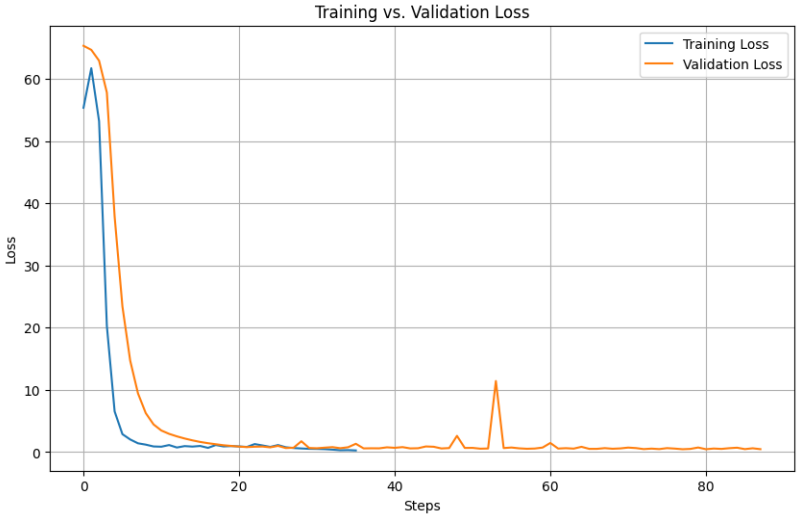


Figure 5.8: Training vs validation loss for AI BHARAT

This graph represents the training and validation losses of a model over time or across a certain number of steps. Effective learning is apparent in that both losses start high before quickly reducing. However, at step 55, the validation loss spikes, which may indicate overfitting or a noisy validation set. The training loss continues to mostly stay low, reflecting that it has done a great job learning the data for training.

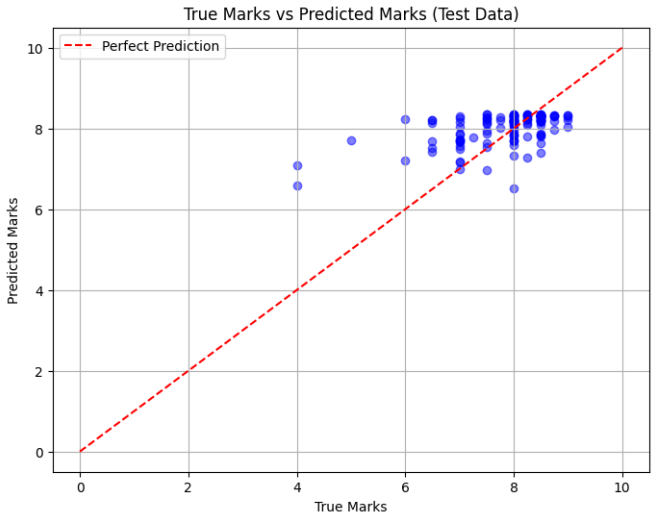


Figure 5.9: True vs predicted for AI BHARAT

In this scatter figure, real and predicted marks are contrasted against each other from the test data. The red dashed line is a perfect prediction where the anticipated

and true marks would be equal. Most dots clustered close to the red line show that the majority of the predictions are accurate. There are a few departures, most of which concern relatively lower true marks, hinting at slight errors in the model's predictions. Overall, the model is doing fine: Forecasts and actual values roughly match.

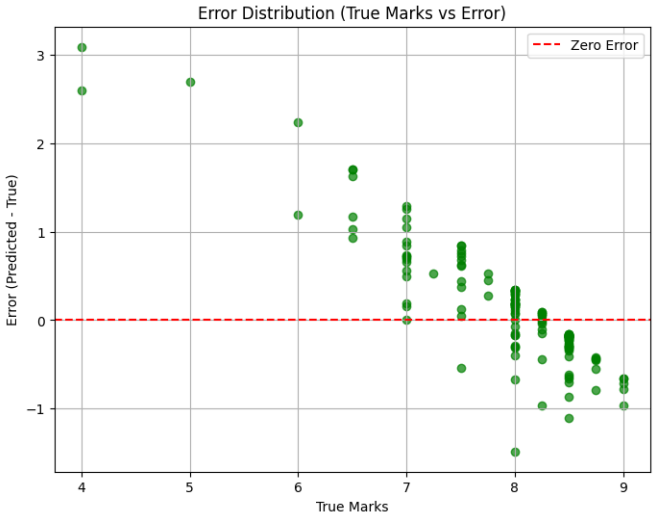


Figure 5.10: Absolute error in prediction for AI BHARAT

This scatter plot represents the difference in the distribution of error between real and predicted marks. It shows the actual marks on the x-axis, while the error that represents predicted - true appears on the y-axis. This red dashed line shows zero inaccuracy at 0. Data points above the line show over-predictions, and below the line, it shows under-predictions. This is evident from the larger variances for lower true marks, reflecting variability in the prediction accuracy for lower values, whereas mistakes are less for higher true marks, which are closer to zero.

5.4.3 MURIL:

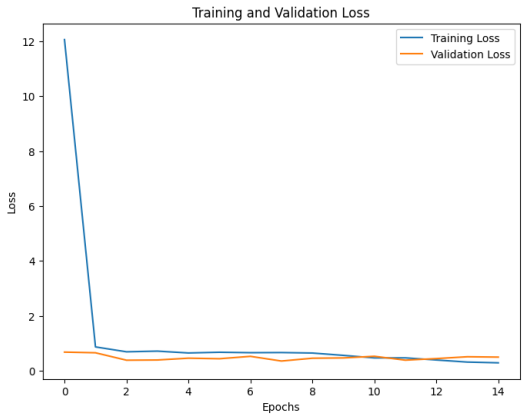


Figure 5.11: Training vs validation loss for MURIL

The graph below shows the training versus validation loss versus epochs. The training loss drops drastically in the beginning, then stabilizes. The validation loss

is rather stable with minor fluctuations, hence there is effective learning without severe overfitting.

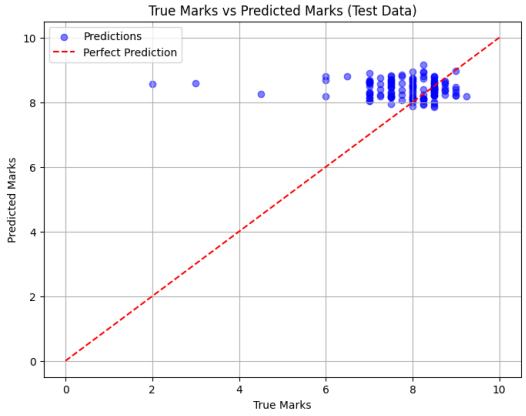


Figure 5.12: True vs predicted for MURIL

The scatter plot showing the actual vs model-predicted scores. The graph has been named as "True Marks vs Predicted Marks (Test Data)". The x-axis is measured as True Marks; the y-axis has values of Predicted Marks. Individual instances in this data are shown with the blue colored scatter points corresponding to the closeness between the model-predicted values and the actual ones. This is the ideal case of predictions-a red dashed line that perfectly fits the true mark, or y=xy = x. Most predictions are around higher true marks, approximately 7-9, with a number of outliers in lower values. It would hence seem that the model is very good for higher scores but can't be trusted as well when it comes to lower scores. This chart helps assess model accuracy by highlighting points that deviate from the line of perfect prediction.

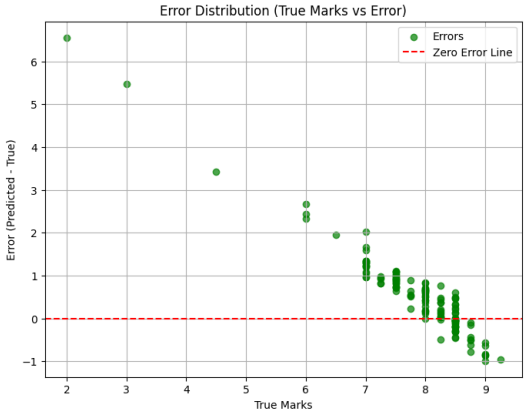


Figure 5.13: Absolute error in prediction for MURIL

5.4.4 Llama 3.1 8B

The scatter plot shows the true marks vs. predicted marks for the test data, with a red dashed line showing the perfect prediction line (y = x). Ideally, all points should align with this line, but deviations highlight prediction errors. In the higher mark range, 7–9, there is high and dense clustering of points, indicating that the model

performs well in this region. However, for lower true marks (2–5), the predictions tend to be high, indicating overestimation in this range. The dispersion of points above and below the line of perfect prediction would appear to indicate that the model suffers from bias and variance issues in particular with respect to the handling of lower-scoring responses. Although this model shows strong predictive accuracy for higher scores, it needs further fine-tuning to improve its generalization for lower mark distributions.

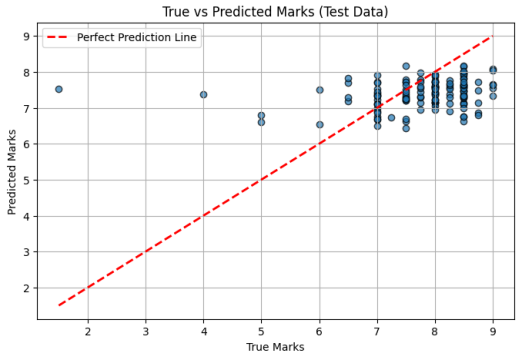


Figure 5.14: True vs predicted for Llama 3.1 8B

The scatter plot visualizes the absolute prediction error against true marks. For low true marks, the error is high; it goes up to approximately 6 for true marks near 1. For the increasing true marks, the absolute error decreases, generally congregating below 1 for true marks between 7 and 9. There is still a bit of variation; some are around 1.5 or 2 points away from their exact value. It shows that, with higher marks, the model performs better, but with lower marks, there are high errors in prediction. This may suggest some bias within the model towards higher marks, with better precision, whereas at low values, the model struggles.

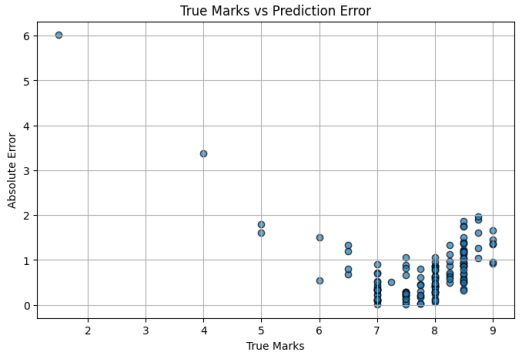


Figure 5.15: Absolute error in prediction for Llama 3.1 8B

5.4.5 Falcon 7B Instruct

This graph plots the Training vs. Validation Loss over 5 epochs,where the x-axis is the epoch, and the y-axis is the loss value. The line in blue color with circular markers shows the training loss, whereas the validation loss is shown by the line in orange color with square markers. The training loss in Epoch 1 is about 0.81 while that of validation loss is about 0.54. This reflects a sharp decline in training

loss by Epoch 2 to 0.22, while the validation loss has remained relatively stable at 0.52. In Epoch 3, the training loss has stabilized at 0.21, with the validation loss decreasing slightly at 0.51. During Epoch 4, the training loss remained the same at 0.21, while the validation loss continued its gradual decline to 0.49. By Epoch 5, the training loss has decreased slightly to 0.20, while the validation loss was at 0.47. This indicates that the model is learning fine, but from Epoch 2 onwards there is a huge difference between training and validation loss, indicating overfitting: while the training loss converges really fast, the validation loss decreases more slowly, which may indicate memorization rather than generalization. To prevent overfitting, one could use dropout, weight decay, or early stopping.

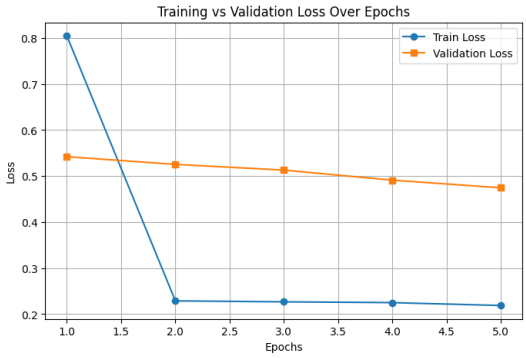


Figure 5.16: Training vs validation loss for Falcon 7B

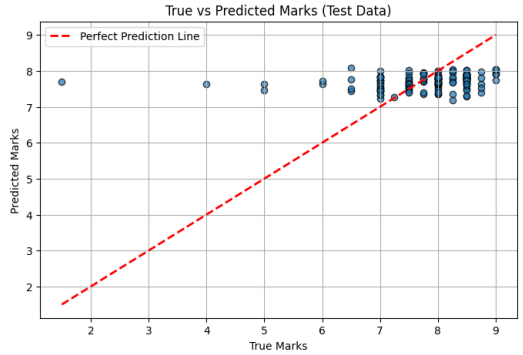


Figure 5.17: True vs predicted for Falcon 7B

This is the graph for True vs. Predicted Marks in the test dataset. The x-axis corresponds to the true marks, and the y-axis represents the predicted marks. The red dashed line represents a perfect prediction line; this is where an ideal model's predictions would lie. Actual model predictions are shown as blue circular points in the scatter plot. Theoretically, all points should ideally fall on the perfect prediction line, which is not strictly the case in this graph. The predicted marks are much higher for lower true marks, about 1 to 5; this represents overestimation by the model. For the higher true marks around 6 to 9, most predictions cluster around the 7 to 8, suggesting that higher values are underestimated by the model. The general trend is positively correlated, meaning that the model is learning something. However, there are problems with extreme values. The dispersion of points around the red line gives the dispersion on the prediction; bias toward mid-range values might also suggest that this model lacks variance in predictions. In order to improve, one may

think of techniques that include error analysis, handling outliers, or taking a more complex model. Moreover, class imbalance or feature scaling issues might be checked to improve predictive accuracy.

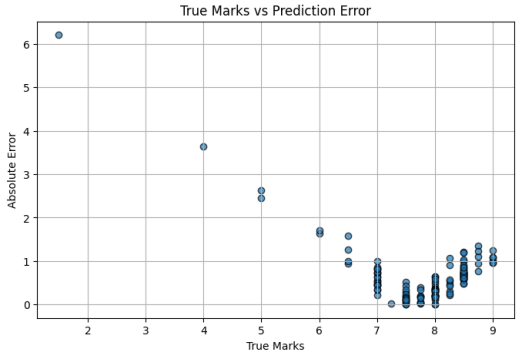


Figure 5.18: Absolute error in prediction for Falcon 7B

This graph shows True Marks vs. Prediction Error, where the x-axis is true marks and the y-axis is the absolute error between true and predicted values. The scatter plot depicts how the model's prediction error varies across different true marks. For lower true marks of about 1 to 5, the absolute error is decidedly high, with some points surpassing 6. That would mean the model struggles with lower values, thus resulting in large overestimation errors. For larger true marks, say over 6, the absolute error generally decreases, with most of the errors being in less than 1, which means better predictions. However, for higher true marks, that is, 7 to 9, there is still some variation in noticeable error but quite small compared to lower marks. The pattern shows that the model is biased toward mid-range values, performing well in higher marks and poorly in lower marks. For better prediction, the approach could be data normalization, handling outliers, or rebalancing the dataset. Additionally, model underfitting in the lower range of marks should also be checked to reduce huge errors in that region.

5.4.6 Falcon 1B Instruct

This graph represents the training and validation loss during five epochs. The training loss starts off with a value of approximately 3.5 for the first epoch and rapidly converges to 0.2 by the second epoch, mostly remaining the same afterward. Whereas the validation loss remains steady with a value of approximately 0.45 in all the epochs. That would mean, though it has optimized rather quickly on the training data, it performs exactly the same on the validation data, which indicates an overfitting problem or a general ceiling for its performance. Plotted points confirm that there is sharp convergence in training loss but not an improvement in validation loss across epochs.

The scatter plot compares true marks versus predicted marks for test data. A red dashed line is drawn on a perfect prediction line where true and predicted values are matching. Most of the predictions are around higher true marks (7-9) with some deviations. The predictions overestimate the lower true marks and show slight variance in the higher values of. This spread around the perfect prediction line does indeed indicate small prediction errors and, therefore, leaves room for improvement in model accuracy. The overall trend does follow the expected diagonal

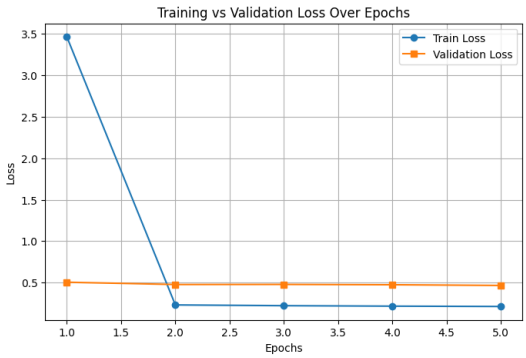


Figure 5.19: Training vs validation loss for Falcon 1B

pattern; however, inconsistencies suggest that there may be biases or limitations in generalization across all score ranges.

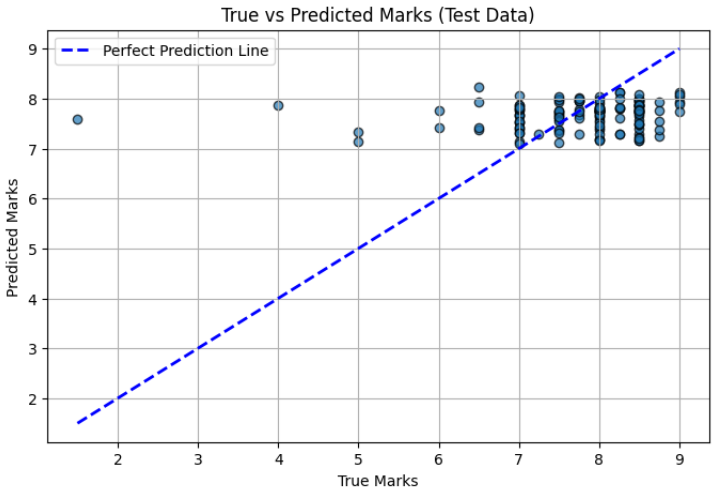


Figure 5.20: True vs predicted for Falcon 1B

This scatter plot shows the absolute error between true marks and predicted values, depicting variation across the score ranges. For lower true marks, say 2–5, there is a higher absolute error, which reaches as high as 6. This suggests significant deviance in the predictions. Above a true mark of 6, the errors tend to be lower, indicating that the model performs better in higher score ranges. The densest clustering of the points is between 7 and 9, with the absolute errors mostly below 1, which shows strong predictive accuracy in this range. The trend in the graph would therefore suggest a bias in the model such that predictions for lower scores are not as reliable, while higher scores tend to give more consistent and accurate estimates. This pattern underlines the need for further model adjustments to improve prediction reliability, in particular, within the lower mark range

5.4.7 Why LLMs Fall Short in AES Compared to Traditional Encoder Models

LLMs may fall short in Automated Essay Scoring (AES) compared to traditional transformer models for several reasons, particularly when working with languages

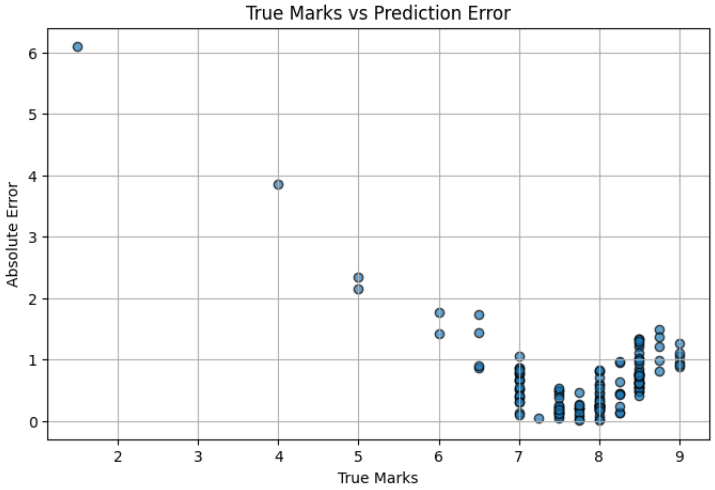


Figure 5.21: Absolute error in prediction for Falcon 1B

like Bangla. One key challenge is that LLMs, despite their vast scale, require extensive fine-tuning on specific language datasets to perform well in niche tasks like AES. These models are specifically constructed for text-generations. And these essay evaluation tasks are especially difficult for languages with less available training data, such as Bangla, where the models may struggle to capture linguistic nuances, cultural context, and syntactic differences that influence scoring accuracy. Traditional Encoder Based Transformer models, on the other hand, are more effectively suited to the particular task at hand, even if they are usually smaller than LLMs. Larger LLMs demand a lot of data, which our models can't handle. Instead, they may concentrate on learning and comprehending the structural elements of Bangla essays, such as syntax, coherence, and relevance to the challenge. The Base Transformer Models are also typically less expensive and computationally more efficient. In contrast, due to the limited institutional resources, we found it challenging to use LLMs efficiently due to their high cost, time commitment, and requirement for high-performance computing resources (strong GPUs and optimized hardware) during training. Additionally, LLMs' size and complexity may make it more difficult to optimize and fine-tune them for AES tasks, particularly when domain-specific or limited data is available, which could result in less-than-ideal performance in comparison. To tackle that issue, we made use of the 4-bit quantization method to fit the model within our 16 GB VRAM. This could slightly sway away the accuracy but not much as the modern quantization methods succeed in keeping the precision fairly well.

Chapter 6

Limitations

Though this study is of great value for automated essay scoring in the Bangla language, there are certain limitations that could have affected the scope and results of the research. The area is continuously developing and new challenges are coming to light. Some of the limitations are mentioned below:

- **Limited availability of Bangla-specific data:** There are a few annotated high-quality datasets for Bangla essays. This makes the diversity and size of the training data limited, probably affecting the generalization capability of the model on unseen or real-world data. The dataset mainly used focuses on a particular essay structure and context, which may not be representative of all Bangla essay styles.
- **Language-Specific Challenges:** Bangla being a morphologically rich language complicates the handling of complex grammar, inflections, and word-form variations not captured by pre-trained models like BERT. The study excluded code-mixing, Bangla-English, and the use of non-standard Bangla script such as Banglish transliterations; therefore, it could only be applied to very informal environments.
- **Bias in Pre-trained Models:** Most models, such as BERT, LLaMA, Falcon, and Bloom, are pre-trained on multilingual corpora that represent Bangla poorly. They might carry certain biases or limitations induced in the pre-training stage. Bangla content in multilingual models is usually underrepresented, which may cause suboptimal performance compared to English-focused essay-scoring systems.
- **Limitations in Model Scalability:** For running larger models like LLaMA 8B or Falcon 7B on a 4080 Super GPU, 4-bit quantization and optimization strategies were necessary. While these techniques enabled the models to run, they may have slightly impacted the performance by reducing precision. Training larger models with extensive Bangla corpora was not feasible due to hardware and resource constraints, limiting the exploration of deeper architectures or custom Bangla models.
- **Computational Trade-offs:** These normally consisted of very resource-heavy models, which needed to be compromised on training-for instance, by using fewer epochs or smaller batch sizes-which could have hindered the model from being at its full potential.

Chapter 7

Future Work

While this research illustrates promising results for the selected models that includes both the Traditional Encoder based Transformer Models as well as the Large Language Models for the automated essay scoring in Bangla, our future perspectives would be to develop the models even further for more accurate predictions. Our initial hypothesis was that the Large Language Models would perform poorly for the purpose of essay evaluations. But to our surprise, they did perform fairly well but not as precisely as that of the BERT based models. As the field of LLM is fairly new, there will still be more future research in this field. So there is more potential that needs to be unfolded regarding the automated essay evaluations using the Large Language Models (LLMs). In order to achieve that, our forefront motive would be to expand our dataset which would include a more diverse range of Bangla data with diverse writing styles. Also incorporating advanced techniques for fine-tuning and transfer learning to enhance and improve our models to yield better results is a significant part of our ambition enabling us to make a more competent Bangla essay scoring system that assists in making the lives of the teachers easier while also making a substantial impact in the AES field.

Chapter 8

Conclusion

The paper inspects AES techniques for the Bangla language and highlights how these techniques may improve Bangla automated assessment scoring processes. AES techniques can accelerate the grading procedure and provide insights into the linguistic aspects of assessments written in Bangla through advanced technologies. The study emphasizes the importance of employing language-specific strategies to address concerns where feature engineering and data scarcity are present in Bangla AES systems. Incorporating Transformer models and Large Language Models (LLMs) into AES systems can enhance performance by enabling better contextual understanding and linguistic analysis, even in resource-constrained settings. Creating some standards for performance evaluation can boost confidence in automated essay scoring. This study adds to the increasing compilation of analysis on Bangla AES by providing perceptions into methodology, difficulties, and prospects for future improvement.

Bibliography

[1] AI4Bharat. *Models by AI4Bharat*. https://models.ai4bharat.org/. AI4Bharat. 2021.

[2] Jay Alammar. *The Illustrated BERT, ELMo, and Co. (How NLP Cracked Transfer Learning)*. URL: https://jalammar.github.io/illustrated-bert.

[3] Y. Attali and J. Burstein. "Automated essay scoring with e-rater® V. 2". In: *The Journal of Technology, Learning and Assessment* 4.3 (2006).

[4] J. Burstein and M. Chodorow. "Automated essay scoring for nonnative English speakers". In: *Computer Mediated Language Assessment and Evaluation in Natural Language Processing*. 1999.

[5] H. Chen and B. He. "Automated essay scoring by maximizing human-machine agreement". In: *Proceedings of EMNLP*. 2013, pp. 1741–1752.

[6] S. Dikli. "An Overview of Automated Scoring of Essays". In: *The Journal of Technology, Learning and Assessment* 5.1 (2006). URL: https://ejournals.bc.edu/index.php/jtla/article/view/1640.

[7] S. Dikli. "The Nature of Automated Essay Scoring Feedback". In: *CALICO Journal* 28.1 (2010), pp. 99–134. URL: https://www.jstor.org/stable/calicojournal.28.1.99.

[8] Data Science Dojo. *Choosing the best llama model: Llama 3 vs 3.1 vs 3.2*. Jan. 2025. URL: https://datasciencedojo.com/blog/llama-model-debate/.

[9] J. Fiacco, D. Adamson, and C. Rose. "Towards extracting and understanding the implicit rubrics of transformer based automatic essay scoring models". In: *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*. July 2023, pp. 232–241.

[10] GeeksforGeeks. *Falcon LLM: Comprehensive guide*. Mar. 2024. URL: https://www.geeksforgeeks.org/falcon-llm-comprehensive-guide/#what-is-falcon-llm.

[11] Google. *MuRIL Large Cased*. https://huggingface.co/google/muril-large-cased. Hugging Face. 2021.

[12] M. G. Hussain et al. "Assessment of Bangla Descriptive Answer Script Digitally". In: *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*. Dhaka, Bangladesh, 2019, pp. 1–5. DOI: 10.1109/ICBSLP47725.2019.202042.

[13] T. Ishida et al. "Large Language Models as Partners in Student Essay Evaluation". In: *arXiv preprint* arXiv:2405.18632 (2024).

[14] M. M. Islam and A. L. Hoque. "Automated Bangla essay scoring system: ABESS". In: *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*. IEEE, May 2013, pp. 1–5.

[15] Y. J. Jong, Y. J. Kim, and O. C. Ri. "Improving Performance of Automated Essay Scoring by using back-translation essays and adjusted scores". In: *Mathematical Problems in Engineering* (2022).

[16] V. S. Kumar and D. Boulanger. "Automated essay scoring and the deep learning black box: How are rubric scores determined?" In: *International Journal of Artificial Intelligence in Education* 31 (2021), pp. 538–584.

[17] A. Kundu and D. Barbosa. "Are Large Language Models Good Essay Graders?" In: *arXiv preprint* arXiv:2409.13120 (2024).

[18] G. G. Lee et al. "Applying large language models and chain-of-thought for automatic scoring". In: *Computers and Education: Artificial Intelligence* (2024), p. 100213.

[19] Jiawei Liu, Yang Xu, and Yaguang Zhu. "Title of the paper". In: *Fopure (Hangzhou) Technology Co., Ltd., Hangzhou, China* (2024). School of Computer Science and Technology, University of Science and Technology of China, Hefei, China.

[20] N. Madnani et al. "Building better open-source tools to support fairness in automated scoring". In: *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*. Apr. 2017, pp. 41–52.

[21] R. Mahjabin et al. "Automated essay scoring for Bangla". In: (2024). Doctoral dissertation, Brac University. URL: http://hdl.handle.net/10361/22776.

[22] K. P. K. Mantha. *Quantization is what you should understand if you want to run LLMs in local environments*. Aug. 2023. URL: https://www.linkedin.com/pulse/quantization-what-you-should-understand-want-run-llms-pavan-mantha.

[23] D. Ramesh and S. K. Sanampudi. "An automated essay scoring systems: a systematic literature review". In: *Artificial Intelligence Review* 55 (2022), pp. 2495–2527. DOI: 10.1007/s10462-021-10068-2.

[24] B. Riordan et al. "Investigating neural architectures for short answer scoring". In: *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Sept. 2017, pp. 159–168.

[25] Sagor Sarker. *Bangla BERT Base*. https://huggingface.co/sagorsarker/bangla-bert-base. Hugging Face. 2021.

[26] M. D. Shermis, J. Burstein, and S. A. Bursky. "Introduction to automated essay evaluation". In: *Handbook of Automated Essay Evaluation*. Routledge, 2013, pp. 1–15.

[27] S. Singh et al. "H-AES: towards automated essay scoring for Hindi". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 13. June 2023, pp. 15955–15963.

[28] K. Taghipour and H. T. Ng. "A neural approach to automated essay scoring". In: *Proceedings of the 2016 conference on empirical methods in natural language processing*. Nov. 2016, pp. 1882–1891.