

# Automated Essay Scoring for Bangla

by

Rubayet Mahjabin

20101011

Shaoli Farzana

20101553

Nishat Zerín

23341114

Sameer Sadman Chowdhury

23341118

Ishmam Hossain

20101145

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  
January 2024

© 2024. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is 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:**

---

RUBAYET MAHJABIN  
20101011

---

SHAOLI FARZANA  
20101553

---

NISHAT ZERIN  
23341114

---

SAMEER SADMAN CHOWDHURY  
23341118

---

ISHMAM HOSSAIN  
20101145

# Approval

The thesis titled “Automated Essay Scoring for Bangla” submitted by

1. Rubayet Mahjabin (20101011)
2. Shaoli Farzana (20101553)
3. Nishat Zerine (23341114)
4. Sameer Sadman Chowdhury (23341118)
5. Ishmam Hossain (20101145)

Of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 22, 2024.

## Examining Committee:

Supervisor:  
(Member)

---

Dr. Farig Yousuf Sadeque  
Assistant Professor  
Computer Science and Engineering  
Brac University

Program Coordinator:  
(Member)

---

Md. Golam Rabiul Alam  
Professor  
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**

Grading papers is one of the most basic everyday tasks carried out in various manners, but the element of complexity always manages to find its way. It is a rigorous task to grade hundreds of papers. Still, the concept of automation has made the job easier as the process decreases the risk of error in checking the papers while simplifying the lives of the teachers. Now, in the case of the English language, this simpleness has already been achieved. However, reaching an equivalent level of sophistication in the context of grading papers in Bangla is still an ongoing process. A team from BUET has researched this very topic in Bangla, but the tools required for grading a paper in Bangla are still far from reaching a distinctive platform. In our research, we have collected datasets containing versatile content to build a competent database and have analyzed the requirements teachers used to grade a paper using natural language processing (NLP) tools. After listing the criteria, we fine-tuned a model using deep learning, in accordance with the criteria to grade a paper written in Bangla with enough accuracy to be considered as relevant as having the same paper graded manually by a professor or a faculty. Our goal is to use transformer models, and embedding along with NLP techniques to grade the essays more precisely, to achieve an industry-standard state-of-the-art system for the Bangla Essay Scoring System.

**Keywords:** Natural Language Processing, Bangla, Grading papers

## **Acknowledgement**

Firstly, all praise to the Great Allah for whom our thesis has been completed without any major interruption.

Secondly, to our supervisor Dr. Farig Yousuf Sadeque sir for his kind support and advice in our work. He has aided us to the best of his abilities.

And finally to our parents for their thorough support without which this would not be possible. With their kind support and prayer, we are now on the verge of our graduation.

# Table of Contents

<b>Declaration</b>	<b>i</b>
<b>Approval</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgment</b>	<b>iv</b>
<b>Table of Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>viii</b>
<b>Nomenclature</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement . . . . .	2
1.2 Research Objectives . . . . .	3
<b>2 Literature Review</b>	<b>4</b>
<b>3 Methodology</b>	<b>10</b>
3.1 Description of Data . . . . .	10
3.2 Preliminary Analysis . . . . .	10
3.2.1 Pre-processing and Data Cleaning . . . . .	10
3.2.2 Tokenization . . . . .	11

3.2.3	Data Cleaning . . . . .	12
3.2.4	Data Splitting . . . . .	12
3.2.5	Exploratory Data Analysis . . . . .	12
3.3	Feature Engineering . . . . .	17
3.3.1	Criteria Problem . . . . .	17
3.3.2	Identification of Criteria . . . . .	18
3.4	Model . . . . .	27
3.4.1	Why BERT Model? . . . . .	27
3.4.2	Why not other models? . . . . .	27
3.4.3	Why Pre-trained Bangla Models? . . . . .	27
3.4.4	Pre-trained BERT Model . . . . .	27
3.4.5	Pre-trained BERT Model (Fine-tuned) . . . . .	29
3.4.6	Pre-trained ALBERT Model . . . . .	30
<b>4</b>	<b>Result Analysis</b>	<b>31</b>
4.1	Model Result . . . . .	31
4.2	Comparison among Previous Researches . . . . .	33
4.3	Limitation . . . . .	35
<b>5</b>	<b>Future Work</b>	<b>36</b>
<b>6</b>	<b>Conclusion</b>	<b>37</b>
	<b>Bibliography</b>	<b>39</b>

# List of Figures

3.1	Pre-Processing . . . . .	11
3.2	Implementation of basic tokenizer, NLTK and Bengali SentencePiece	11
3.3	Scatter Plot of word count and marks . . . . .	13
3.4	Graph of cosine similarity and question answer pair . . . . .	14
3.5	Scatter Plot of word count and marks (BGS) . . . . .	15
3.6	Graph of cosine similarity and question answer pair (BGS) . . . . .	16
3.7	Scatter Plot of word count and marks (SSC) . . . . .	16
3.8	Graph of cosine similarity and question answer pair (SSC) . . . . .	17
3.9	Spelling Criteria . . . . .	18
3.10	Grammar Criteria . . . . .	18
3.11	NER tags . . . . .	19
3.12	Example of POS Tagging . . . . .	21
3.13	Result of Grammar check using translation . . . . .	22
3.14	Working Grammar Tool . . . . .	23
3.15	BERT Model . . . . .	28
3.16	Fine-Tuned BERT Model . . . . .	29



# List of Tables

3.1	Creative QA Marks Distribution . . . . .	18
3.2	BNG103 Marks Distribution Provided By Faculty . . . . .	18
4.1	Essay Scoring Evaluation . . . . .	31
4.2	Creative QA Score Evaluation (Test) . . . . .	31
4.3	Creative QA Score Evaluation (Validation) . . . . .	31

# Nomenclature

The upcoming list elaborates on various symbols & abbreviations which will be utilized later in the body segment of the document

*AEG* Automated Essay Grading

*BERT* Bidirectional Encoder Representations from Transformers

*BOSWE* bag-of-super-word embeddings

*CVA* content vector analysis

*ETS* Educational Testing Service

*GLSA* Generalized Latent Semantic Analysis

*LDA* latent Dirichlet allocation

*LSA* Latent Semantic Analysis

*NLP* Natural Language Processing

*PLSA* Probabilistic Latent Semantic Analysis

*QWK* Quadratic weighted Kappa

# Chapter 1

## Introduction

Evaluating students' writing is one of the educational system's most expensive and time-consuming responsibilities, as assessing student work and giving them insightful feedback takes time. Accuracy being a crucial factor for the Automated Essay Grading (AEG) system, could be the answer to this problem. AEG systems are developed to compute similar results to human raters and place significant focus on spelling, grammar, punctuation, and simple stylistic properties, but these properties are not sufficient enough to measure the quality of essays.

Automated Essay Grading (AEG) systems have been an enthralling problem in the world of Artificial Intelligence (AI) and Natural Language Processing (NLP). Due to this, there have been advances to solve this problem, such as recurrent neural networks, regression methods, Latent Semantic Analysis (LSA), Hybrid systems, and many more, but all these methods were employed on AEG systems of the English language. Whereas, Bangla is the 5th most spoken language in the world and there hasn't been sufficient development for solving this problem.

There has been research on AEG in Bangla using Generalized Latent Semantic Analysis (GLSA), and these have provided significant results. Matters like proximity problems removing stop words, removing grammatical errors, word stemming, etc. were taken into consideration while creating the grading system but there are other aspects that need to be addressed. In addition to these aspects, our Bangla grading system will go through better dataset training. With this, the system can achieve a high accuracy rate [7].

The recent work on Bidirectional Encoder Representations from Transformers has appeared to have the potential for solving the in-depth contextualization problems in essay scoring as the model is trained to various tasks such as QA, and named entity recognition [13]. In the case of Bangla, there have been innovations in the pre-trained Bert Model in recent times showing immense results in sentiment analysis and classification. Pre-trained models using a large Bangla corpus have the possibilities to be essential in terms of understanding the depth of content and thus grade it accurately.

In this paper, we have looked at past research on automated essay grading systems and identified the existing errors in those approaches. After identification and investigation of the errors, we have applied our new approach to optimize the missing factors from the current workings. Here, we have used our own diverse dataset to test the approach and its usage in different Bangla content. Finally, we analyzed the outcome through testing to ensure further advances and presented our findings regarding automated essay grading systems for Bangla.

## 1.1 Problem Statement

Automated Essay Grading (AEG) is important because it speeds up the procedure and requires less work from human raters to grade the essays as closely as possible to their judgments. There have been great works on AEG based on the English language over time, and more research on Bangla Automated Essay Grading Systems should be done as well.

In a research paper [7], there has been previous work on AEG using Generalized Latent Semantic Analysis (GLSA). Still, there are other methods that haven't been explored in the case of automated essay grading systems for Bangla. In addition, there were only 3 sources for data, which were two titled essays where 100 scripts were used for pre-training and S.S.C level Bangla literature where 80 scripts were for training, which is insufficient to create a robust dataset for a language that is largely diverse. Also, Bangla is a versatile language with many aspects that haven't been solved previously. With the Bangla NLP community growing and essay scoring consisting of multiple aspects, we believe we can acquire better results using the latest techniques of NLP to create a better AEG system for Bangla.

In another research paper [16], the established work is catered to the Education System of Bangladesh. The work emphasized assessing various prospects in English Essay scoring techniques and dives into techniques that can be implemented to grade Bangla script using linguistic analysis and machine learning. However, the insufficiency is still visible as it was tested on 20 answer scripts, pointing out that various aspects deserve detailed research and investigation.

As research on Bangla AEG is being published, different issues are being displayed such as insufficient and unauthentic datasets, lack of contextual depth in terms of grading, and many more. Thus, we want to focus on these issues using developed NLP techniques and previous works on Bangla essays, and creative question-answers.

So, now the question arises, **Can we create an Automated Essay Grading System based on different marking criteria using the Transformer model?**

We will be answering this question throughout our research plan and evaluating it using natural language processing and machine learning techniques.

## 1.2 Research Objectives

Our research intends to explore different Automated Essay Grading (AEG) approaches to solve various problems existing in the present Bangla AEG system. So, for this purpose, the objectives in the plan are:

1. To build a standard and realistic dataset for the Bangla Essay Grading System.
2. To evaluate the requirements of grading Bangla papers according to certified teachers.
3. To understand techniques such as tokenization, stopword removal, lemmatization, and many more holding on the Bangla Linguistic aspects.
4. To test and improve tools such as evaluation metrics such as spelling corpus, transformers, etc.
5. To develop, apply, and evaluate an AEG-based model on Bangla Bert Base and Albert.
6. To provide further propositions on upgrading the model by training different types of databases.
7. To disseminate our research via conference and journal papers.

# Chapter 2

## Literature Review

The present limitations may hinder the automation of Bangla AEG system development but the paper “Automated Essay Scoring Using Generalized Latent Semantic Analysis” [7] by Md. Monjurul Islam puts perspective by employing a method utilizing GLSA which stands for Generalized Latent Semantic Analysis to assess Bangla essays. The model uses a document matrix that consists of n-gram which is different than LSA. The developed system appears to be more accurate and capable of grading essays in Bangla. The system has three main modules: the training trial generation module, the ABESS scoring module, and the performance evaluation module. Pre-processing was done in the following stages: stopword elimination, words stemming to the roots, and then choosing n-gram index terms. Using synthetic essays, a precision of 95% is obtained. Using the essay submitted by the student, they obtained a precision of 96.25%. But using the descriptive answer, they found low accuracy which is 65%. On average, the system is 89% to 95% accurate in comparison with a physical grading.

Md. Monjurul Islam[8] worked on another paper with the same approach used in his previous paper [7] but the dataset differed. In that research, two sources were mentioned containing essay questions and answers and the notion of true positive achieving 100% and 93.7% accuracy elaborated the work’s accuracy.

Afterward, Hussain et al.’s paper [16], ”Assessment of Bangla Descriptive Answer Script Digitally,” addresses a gap in techniques primarily designed for English assessments. He proposes an automated method for evaluating answer scripts in Bangla. Statistical insights into Bangladesh’s education system are provided and the challenges faced by teachers in assessing a large volume of students are brought to light by this paper. The complexities faced in assessing subjective answers in Bangla are addressed by this research because very few resources are available to help aid this field. The authors discuss various evaluation techniques for English scripts, including machine learning and natural language processing. Their model, involving question and answer classification with keyword matching and linguistic analysis, achieved a minimum relative error of 1.8% on 20 answer scripts. The authors address the simplistic nature of the model and encourage further research so that improvements are made for assessing subjective Bangla scripts efficiently.

The study by Burstein et al. [1] shows that essay features are automatically an-

alyzed based on writing traits listed at each of the six scoring points in the manual scoring guide utilized by human raters. A system is created that might grade an essay using the criteria listed in the manual scoring guide. Examples of the features include topical analysis, syntactic structure, and rhetorical structure. For each essay question, a stepwise linear regression analysis is run on a training set to extract a weighted set of predictive features for each test question. By matching the terms in an essay to those in manually graded instruction, the e-rater assesses its lexical and subject substance. The Electronic Essay Rater (e-rater) score predictions and human rater scores ranged from 87% to 94% across the 15 exam questions, according to the linear regression analysis.

In another paper paper by Attali and Burstein [5] is about the latest version of the e-rater software program, E-rater V.2, that has been developed by the Educational Testing Service (ETS) to automatically grade essays. Measures of grammar, usage, mechanics, style, organization, development, lexical complexity, and prompt-specific vocabulary usage are features existing in the system in order to evaluate essays with accuracy using NLP and machine learning. Criterion has an application to extract feedback using writing analysis tools to get agreement errors, verb formation errors, wrong word use, missing punctuation, and typographical errors. It is proven that e-rater V.2 has more reliability than a human rater due to its functions, and with a human and machine correlation of 0.97, it has more potential than a human rater, even though it might not be perfect.

Before these advancements, a study by Burstein et al. [2] suggests that the long-term objective of automated essay scoring is to be able to provide test-takers or instructors with diagnostic or instructional information in addition to a quantitative result. E-rater employs a hybrid feature approach by combining a number of variables that are calculated statistically or obtained using NLP methods. The training set for this research contained 270 essays in total. After testing, the researchers came to the conclusion that discourse, syntactic, and topical information can be trustfully employed for machine prediction of essay scores. This is because the accuracy of agreement between the e-rater and human rater ranged from 87% to 94%.

Afterward, Burstein, Chodorow, and Leacock [3] developed Criterion, an online writing service with a teacher's scenario to evaluate essays and give feedback specifically. Criterion consists of two applications. One is Critique which alerts about grammatical, usage, and mechanics errors using "bigram" to find occurrences of words expected from the essay and also identifies the discourse structure and undesirable stylistics of the essay. The second application, e-rater 2.0, gives word-based reviews and holistic scores. Features of e-raters represent the value of the essay prominently. In the research, it is seen that 71% of confusable word errors, 92% of subject-verb agreement errors, and 95% of possessive marker errors were accurately found. For the e-rater, the adjacent agreement between e-rater 2.0 and the human score is approximately 97%. Other than technical issues, teachers have given positive feedback throughout the evaluation process.

The AEG system for the Indian language faces the problem of the local Indian language existing in the English essays. So, Ghosh and Fatima [6], have consid-

ered the issue of disruption due to the Indian local language in the automated essay grading system, leading to lower scores through artificial checking. For this purpose, the authors have proposed a framework to identify and solve the problem of local languages' effects on English essays. There will be two parts to the automated essay grading system. The score reporter will recognize the local language and allow the person to replace the word with proper English words. Afterward, the essay will be graded, including information such as the number of local words and the effect of the local words. Whereas, the diagnostic feedback provider collects all the modifications that could be applied to improve the grade like grammatical mistakes, redundancy, usage of weak words, etc.

On the other hand, Robert Ostling et al. [9] approach developing an automated essay scoring system for Swedish, where a corpus is created from 1702 essays from Swedish high schools. In the given approach, essays are graded by two different humans, so the concept of "re-grading" is displayed here. The system will be able to categorize grades into four different types, which are IG, G, VG, and MVG, which means fail, pass, pass with distinction, and excellent, respectively. Three kinds of features: simple, corpus-induced, and language error are present in the system. Text length, average word length, and OVIX lexical diversity measure corporate simple features. The same grading between blind grader and computer is about 57.6% whereas 53.6% is found to have the same grade given by their teacher. In the re-grading process, 8.7% of cases are found to have a one-step difference in grading.

Essay grading requires feature specifications and to implement these features, Natural Language Processing (NLP) and Machine Learning tools can provide assistance such as Cozma, Butnaru, and Ionescu [12], proposed combining string kernels (low-level character n-gram features) with recent word embeddings (high-level semantic features) method known as the bag-of-super-word embeddings (BOSWE) to obtain state-of-the-art AES results. They used the Kaggle Automated Student Assessment Prize (ASAP) 1 dataset to evaluate their strategy. As training data, all essays in the source domain are used. Results show that the histogram intersection string kernel alone achieves better overall performance for the in-domain automatic essay scoring challenge (0.780).

Entities represent the subject and purpose of the essay content so to recognize these entities Named Entity Recognition (NER) has been a technique used in different AEG systems. In the paper "Banner: A Cost-Sensitive Contextualized Model for Bangla Named Entity Recognition" [17], the NER was done on the Bangla language applying Word2Vec and BERT embeddings. The approach deliberately handicaps the dominant class for it to learn at a slow pace while altering the cost-sensitive loss function and layering the Conditional Random Field (CRF). This causes the development of 8% in F1 MUC score using newly presented NER dataset for Bangla.

The paper "Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms" [19] discusses the new approach to sentiment analysis applying deep learning models and rule-based methods BTSC (Bangla Text Sentiment Score) along with (LDD) lexicon data dictionary. Out of all the proposed models, BERT with LSTM has worked best with an accuracy of 84.18%.



Various approaches have been suggested in order to progress the research on AEG systems through time. For example, in the paper by Kakkonen et al. [4], the primary objective when using Probabilistic Latent Semantic Analysis (PLSA) for essay grading is to construct the model from training data. Based on the Spearman correlation between a person's grades and the system's grades, it was found that the accuracy of both approaches was very close. The results of this research showed that LSA provided better results than PLSA and PLSA-C. In contrast to the statement of Hofmann (2001), PLSA performed equally well or even better than LSA in the context of information retrieval. Also, it was seen that combining various similarity scores from models with various numbers of latent variables improved overall accuracy.

Bengali sentences also have been experimented on using the semantic analysis in "Semantic Analysis on Bengali Sentences" [14] by Khatun and Haque which delves into Bangla linguistic characteristics and grammatical attributes and experiments on different types of sentences with length to acquire context on Bangla language.

Whereas, Taghipour and Ng [10], used recurrent neural networks in their research to develop an automated essay grading system. A neural network-based automated essay scoring system makes sure that an essay's score correlates with its performance. It becomes simpler to perform the job of scoring the essay without human touch since it can handle complex writing through non-linear neural layers and enables the system to be free from manual feature engineering. The official evaluation metric in this study is QWK (Quadratic Weighted Kappa), which assigns the essay a score between 0 and 1, and the neural network system design comprises five layers. The analysis reveals that LSTM produces outcomes that are 4.1% better. According to QWK, the best system outperforms the baseline by 5.6%.

Observations also have a significant influence in advancing the progress of developing essay scoring systems. V. V. Ramalingam et al. [15] found large datasets with various patterns can help us achieve better results due to machine learning techniques with several feature spaces. It divides a corpus of textual entities into a few distinct groups corresponding to different grades. Along with numerous other classification and clustering techniques, the model will be trained using the linear regression technique. The evaluation was done by comparing the resulting values with the dataset values after processing the training set through the downloaded dataset. Features extracted from the ASCII text of the essays are word count and sentence count - text mining library, POS Tag - NLTK library, spelling mistakes: spell checker provider named "enchant", and Domain Information Content. The difference between both scores is not much, indicating that even the machine is capable of assessing an essay like a human rater, which can be utilized for practical purposes. The lack of autocorrelation, homoscedasticity, and multicollinearity in the current system are its drawbacks.

Jong, Kim, and R [20] revealed that the Automated Student Assessment Prize dataset is applied to the algorithm in order to extend the number of essay-score pairings via back translation and score modification. In addition to this, the like-

likelihood of data augmentation was demonstrated by modifying the essay’s score. To diversify the enriched data, back-translation essays were created utilizing two languages. Using the supplemented data, the performance of both models was enhanced by an average of 0.2%.

The summary of the most recent advancements in Automated Essay Evaluation (AEE) is visible in Zupanc and Bosnic’s [11] paper which provides a thorough summary of the most recent advancements in Automated Essay Evaluation (AEE). It also provides an overview of the NLP area and the current commercial and openly accessible AEE solutions. Grammar, mechanics (such as spell-checking problems, capitalization errors, and punctuation errors), substance, lexical sophistication, style, organization, and content development are only a few of the attributes used by existing AEE systems to describe essay features. To assess the essay’s semantics, several systems employ Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), and Content Vector Analysis (CVA). One of the most vital problems with AEE is the lack of development of tools for different languages, which happens to be the main aspect of AEE. Hence, language dependency is a very strong disadvantage of AEE, along with other problems such as lack of datasets, lack of consideration of text semantics, and validation of the system.

BERT model has showcased its influence through many works and the start was the paper by Devlin et al. [13], which initialized the BERT Model overviewed obstacles regarding language models to address and improve many NLP tasks such as named entity recognition or QA and emphasizes designing bidirectional transformers in order to handle multiple tasks with the same model architecture while fine-tuning the model according to the task. The model is constructed to predict relation among sentences pre-training and fine-tuning. In pre-training, the model is trained with unlabeled data, and some percentage of the token is masked so that the model can predict the masked tokens and then make the next sentence prediction (NSP). Afterwards, fine-tuning is done with pre-trained parameters. The transfer model has provided significant results such as on the General Language Understanding Evaluation (GLUE), the BERT base and BERT large have overcome all the system on all tasks by the accuracy of 79.6% and 82.1%.

While there is research on various approaches to automated essay grading systems, few researchers have considered the BERT model for essay scoring. According to Kowsher et al., [21], the BERT model is heavily dependent on resourceful language corpus. So, in the perspective of languages, the performance can differ. For this reason, a monolingual BERT model, the Bangla Bert Model was proposed in this paper, catering to pre-training datasets such as BanglaLM. The research revolves around its productivity in sentiment analysis, named entity recognition, binary, and multilevel text classifications. The result shows that it has worked in detecting fake news incorporating an accuracy of 99% and also performing well in sentiment analysis with 97%.

Involving the BERT model, Mayeesha et al., [18] in this paper have researched on question answering systems in the Bangla language using deep learning methods where text-based reading comprehension is focused. A fine-tuned BERT model is

used to pre-train the model with a large reading comprehension dataset using a large subset of SQuAD 2.0 which was translated into Bangla. RoBERT and DistilBERT were also experimented with in order to compare the result with the zero-shot and fine-tuned BERT model. Even though it under-performed for the dataset collected from children of grades three and four, its result was efficient for the translated dataset.

# Chapter 3

## Methodology

### 3.1 Description of Data

The dataset utilized in this research paper is what separates it from the common crowd because of the comprehensive collection process that was carried out to fabricate it. A total of 324 graded scripts from BRAC University Residential (BNG103) were included, providing a substantial corpus for analysis. Further contribution came from the inclusion of 200 copies of class-8, 9, and 10 papers on topics related to Bangladesh and Global Studies (Dhanmondi Govt. Girls' High School). Notably, the dataset encompasses scripts from the SSC first paper (50) from Monsurpur Abdul Hamid Talukder High School, adding a specific educational context to the collection. The dataset contained in a CSV file, named BNG103, and structured with six columns: Name, Answer, Marks, Question, Feedback, and Folder, ensures a comprehensive representation of relevant information for the research inquiry. Another CSV file is structured with three columns: Question, Answer, and Marks, and contains the data collected via the Dhanmondi school and Monsurpur school. The compilation of this unique, one-off dataset is what makes this research so prominent and justifies its validity as a strong contributor to the research of AEG systems.

### 3.2 Preliminary Analysis

#### 3.2.1 Pre-processing and Data Cleaning

In this research, we considered many aspects for pre-processing the original data, as Bangla essays consist of unnecessary information that can't be used to predict scores. In our case, scripts from the course BNG103 of BRAC University consisted of front page, logos, and name in text files which don't fit the criteria for grading a Bangla essay, that is why it is considered an irrelevant parameter.

The dataset contains scripts with student names and it is considered private information. So, for data privacy purposes, a unique identifier named "unique\_id" was assigned instead of the student name which adds a new column "Name" with a unique ID following the below code snippet:

```
data['unique_id'] = range(10001, 10001 + len(data))
```

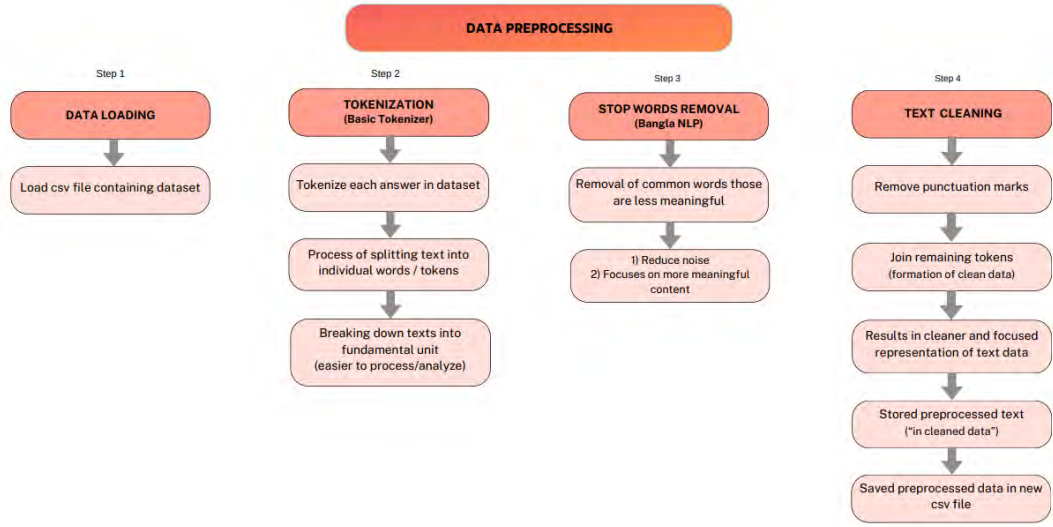


Figure 3.1: Pre-Processing

### 3.2.2 Tokenization

Tokenization was done to divide texts into fundamental units for easier analysis. There were options such as NLTK Tokenization or Bengali SentencePiece Tokenization but they didn't fit the purpose as NLTK is not suitable for the Bangla language and Bengali SentencePiece deconstructs the content in a way that loses its meaning. So, Basic Tokenizer is the accurate tool to tokenize bangla essay scripts among all the options with its simple use and no need for a pre-trained model.

INPUT	OUTPUT		
	BASIC TOKENIZER	NLTK	BENGALI SENTENCE PIECE
কবি রবীন্দ্রনাথ ঠাকুর (১৮৬১-১৯৪১) এর ১৯১৯ সালের পটভূমিতে লেখা উল্লেখযোগ্য ছোটগল্প হচ্ছে “ পোস্টমাস্টার “। রবীন্দ্রনাথ ঠাকুর এর মধ্যবয়স্ক সময়কার “২য় পর্বের” ছোটগল্প থেকে নেওয়া এই ছোটগল্পটি। কবি রবীন্দ্রনাথ ঠাকুর কে ছোটগল্পের জনক বলা হয়। শুধু ছোটগল্প বা কবিতায় নয় তার ভূমিকা কৃষিকাজ, শিক্ষা ব্যবস্থা, চিত্র অঙ্কন সহ নানান বিষয়ে অপরিসীম। বাংলার প্রথম “নোবেল” ১৯১৩ সালে তার হাত ধরেই আশা।	কবি রবীন্দ্রনাথ ঠাকুর ( ১৮৬১ - ১৯৪১ ) এর ১৯১৯ সালের পটভূমিতে লেখা উল্লেখযোগ্য ছোটগল্প হচ্ছে “ পোস্টমাস্টার “ । রবীন্দ্রনাথ ঠাকুর এর মধ্যবয়স্ক সময়কার “ ২য় পর্বের ” ছোটগল্প থেকে নেওয়া এই ছোটগল্পটি । কবি রবীন্দ্রনাথ ঠাকুর কে ছোটগল্পের জনক বলা হয় । শুধু ছোটগল্প বা কবিতায় নয় তার ভূমিকা কৃষিকাজ , শিক্ষা ব্যবস্থা , চিত্র অঙ্কন সহ নানান বিষয়ে অপরিসীম । বাংলার প্রথম “ নোবেল ” ১৯১৩ সালে তার হাত ধরেই আশা ।	কবি রবনদরনথ ঠকর ১৮৬১১৯৪১ এর ১৯১৯ সলর পটভমত লখ উললখযোগ্য ছটগলপ হচছ পসটমসটর রবনদরনথ ঠকর এর মধ্যবয়সক সময়কর ২য় পরবর ছটগলপ থক নওয় এই ছটগলপট কব রবনদরনথ ঠকর ক ছটগলপর জনক বল হয় শখ ছটগলপ ব কবতয় নয় তর ভমক কষকজ শকষ বযবসথ চতর অঙ্কন সহ ননন বয়য় অপরসম বলর পরথম নবল ১৯১৩ সল তর হত ধরই আশ	_কব_রব ন দর ন থ_ঠক_১৮৬১ ১৯৪১_এর_১৯১৯_সল_পট ভ মত_ল থ_উল ল খ য গ য_ছট গল প_হ চ ছ_পস টম স টর_রব ন দর ন থ_ঠক_এর_মধ্য ব য়স ক_সময় কর_২য়_পর বর_ছট গল প_থ ক_ন ও য_এই_ছট গল পট_কব_রব ন দর ন থ_ঠক_ক_ছট গল_জনক_বল_হয়_শ খ_ছট গল প_ব_ক_বত য_নয়_তর_ভ মক_কষ ক জ_শ ক য_ব য বস থ_চ তর_অ ঙ ক ন_সহ_নন ন_ব য় য_অ পর

Figure 3.2: Implementation of basic tokenizer, NLTK and Bengali SentencePiece

### 3.2.3 Data Cleaning

We needed to process different stopwords to remove them to reduce the noise of the text and center the attention toward meaningful content. There were three choices of tools which were NLTK, spaCy, and BanglaNLP. But for automated essay scoring for Bangla, BanglaNLP (BNLP) is proven to be the best tool for removing stop words due to its accuracy, comprehensiveness, and expert curation. While NLTK and spaCy offer stopwords lists for Bangla, BanglaNLP's stopwords list is specifically tailored for Bengali language processing tasks such as grading because it keeps the meaning of the content intact while extracting dispensable parts of the essay.

Bangla Language grammar contains various punctuations which are often regarded as meaningless for the content so removal of punctuation is necessary to create a clean dataset. The work is done using the below algorithm:

```
translator = str.maketrans({'!': None, ',': None, '?': None, '": None, ':': None, "'": None, '_': None, '...': None, '!': None, '(': None, ')': None, '*': None, '"': None, "'": None})
```

Afterward, we merged all the cleaned data of the answer together and formed the clean text to make the data appropriate to train the deep learning models later on for score prediction. To work with pre-processed data, we have loaded it in CSV files. The structure essentially reads CSV files, concatenates them, removes unnecessary columns, adds a unique identifier, performs text cleaning, and saves the processed data to a new CSV file.

### 3.2.4 Data Splitting

The Pandas library and scikitlearn's `train_test_split` function were used to split a combined dataset, loaded from a CSV file ('combined\_cleaned\_data\_tarc.csv'), into three parts: training, validation, and test sets. Furthermore, the `train_test_split` function is employed twice to achieve this division. Initially, it splits the combined dataset into training (`train_df`) and the rest (`rest_df`) using 20% for testing and fixing the random state to 42. Subsequently, the rest of the dataset is further separated into validation and testing using a 50-50 split ratio. Finally, each of these datasets is saved into particular CSV files, excluding the index column to maintain a clean and portable representation of the respective sets to work in training, validation, and testing models.

### 3.2.5 Exploratory Data Analysis

After going through the dataset, we have considered the score, question, and answer. Through this, we can point out the relation between essay word count and marks. A lambda function and the `.apply()` method are used to compute the word count. By applying `str(x).split()`, we split the essay text into words and calculate the length of the resulting list. This is done for each row in the 'Answer' column. If the 'Answer' is not a string (e.g., a float or NaN), the word count is set to 0. A new column,

'essay\_ word\_ count' is created in the 'combined\_ data' DataFrame to store the word count of each essay.

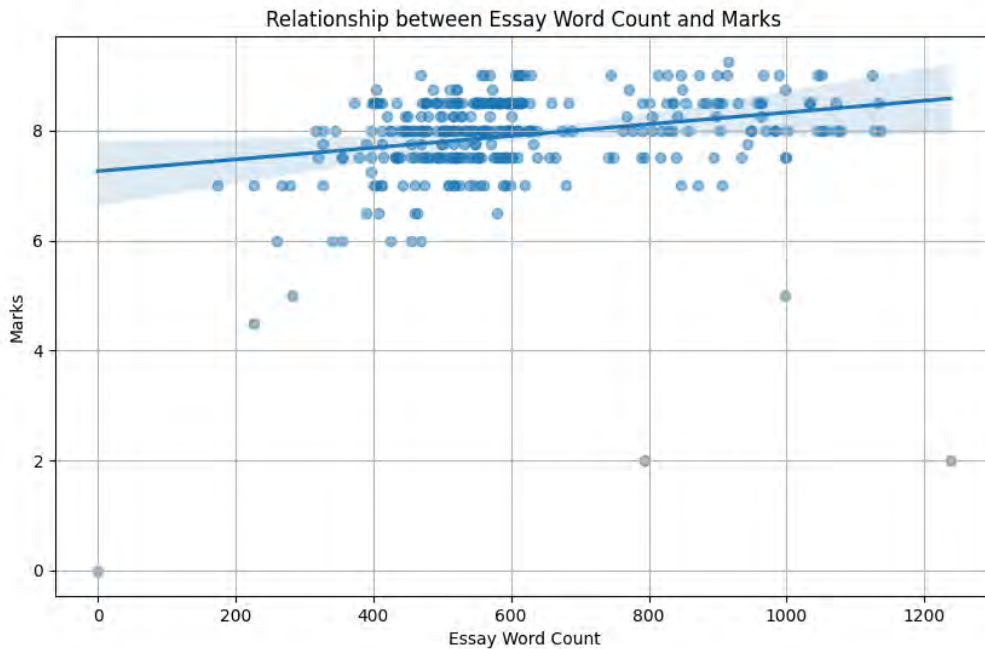


Figure 3.3: Scatter Plot of word count and marks

According to figure 3.3, the increasing word count is positively proportional to the grade, allowing us to consider it as a criterion for score prediction. The scatter plot is clustered around 400-600 words to 800-1000 words where the availability of 400-600 words seems more, and the ratio of high score is also present due to the visibility of the regression line. So, the scatter plot indicates that students who write more have higher chances of high marks.

Conceptual and opinion-based questions require concise and relevant answers. It isn't any different in terms of Bangla writing. In this research, the dataset possesses poem and story-related questions that justify an opinion. The relatability of the answer with the poem and story requires the system to understand the answer. For this reason, the relevance of the question and answer is necessary.

The function `calculate_cosine_similarity` computes the cosine similarity between the question and answer. It checks if the question or answer is missing (NaN) and returns a similarity score of 0.0 in such cases. It tokenizes the text, converts it to lowercase, and calculates TF-IDF vectors.

From figure 3.4, it can be observed that the relevance between question and answer is up to 0.7. If the secondary text has a cosine similarity score closer to 1, it refers to having close relevance with the reference text. It points to the dataset's potential to improve its cosine similarity for better results because the graph is pointed more toward the 0.4-0.5 zone.

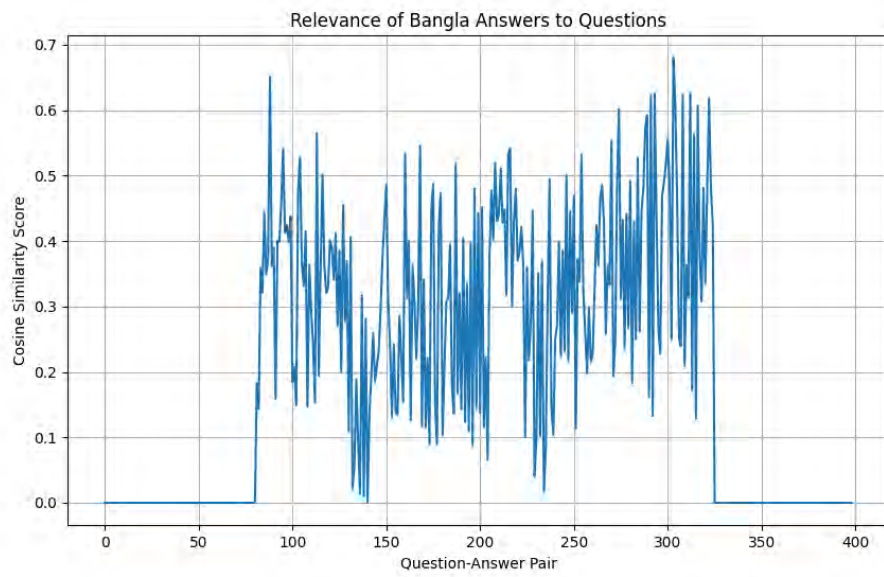


Figure 3.4: Graph of cosine similarity and question answer pair



The creative QA have showcased similar results in preliminary analysis. In the case of BGS scripts,

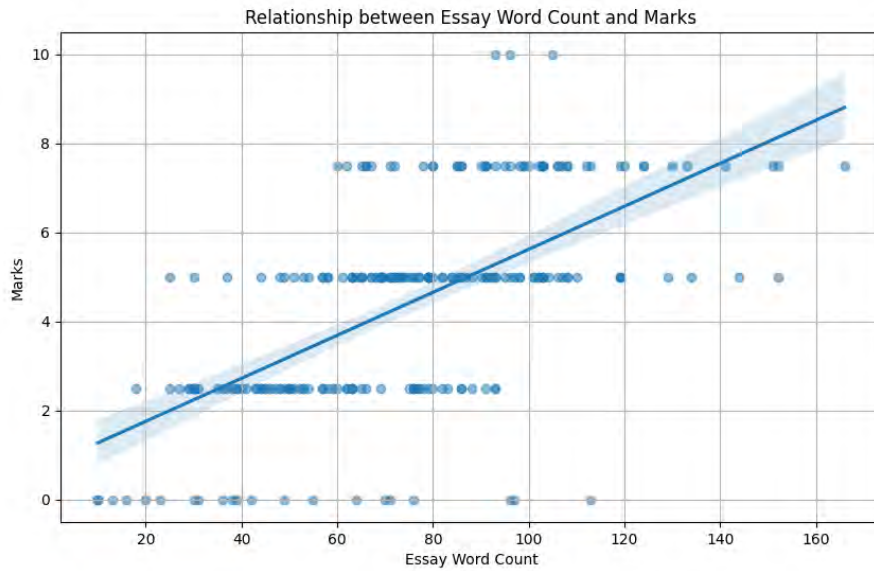


Figure 3.5: Scatter Plot of word count and marks (BGS)

Figure 3.5 indicates that with increasing word count, marks have also increased linearly.

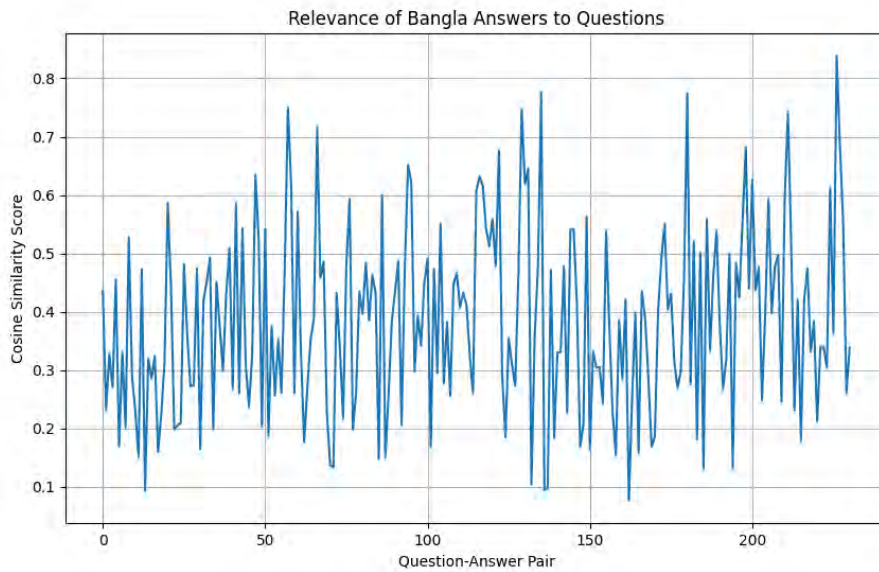


Figure 3.6: Graph of cosine similarity and question answer pair (BGS)

Here in figure 3.6, the cosine-graph relativity is clustered around 0.2-0.6 meaning the student has related their answer to the question around 20-60%. And for SSC scripts,

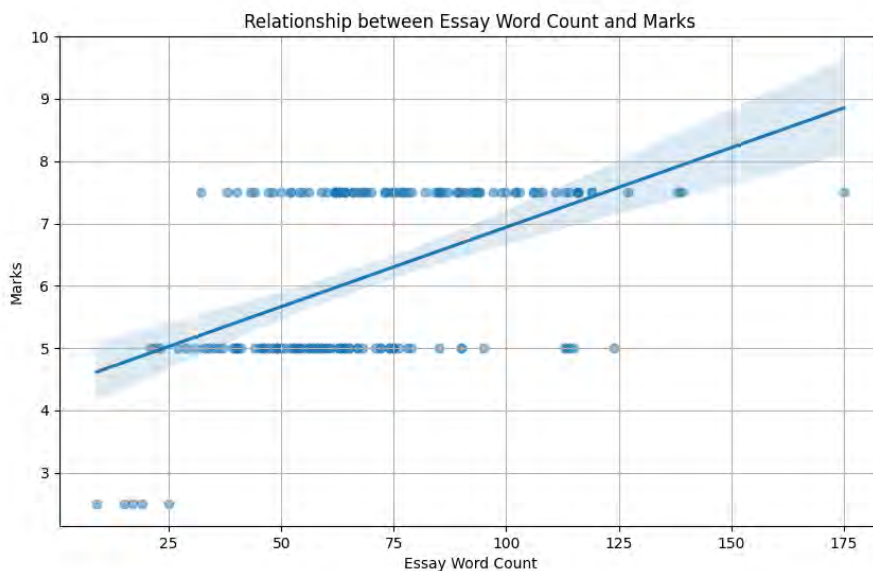


Figure 3.7: Scatter Plot of word count and marks (SSC)

It proves the point again that word count matters in terms of grades but here the marks are around mainly 5 and 7.5. This categorizes the knowledge and capabilities of the students. Our judgment on creating criteria heavily matters on this notion.

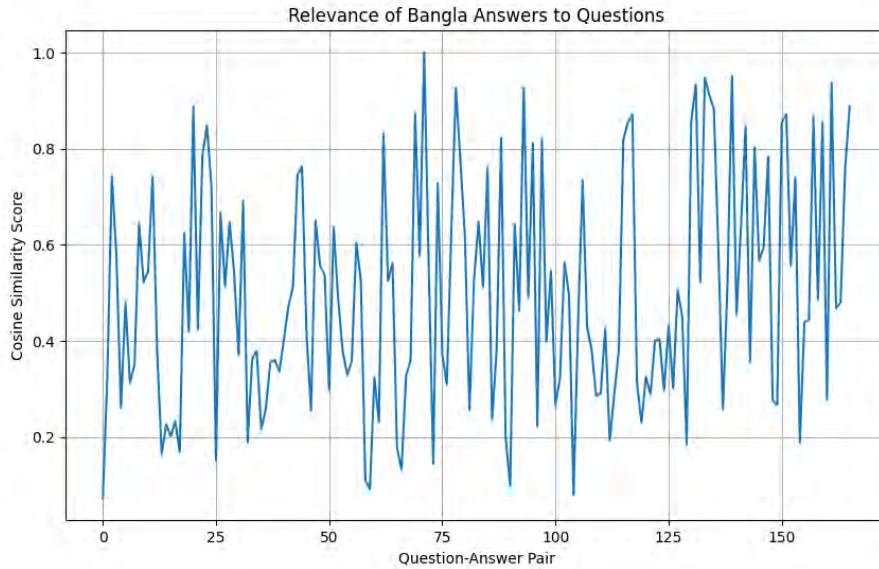


Figure 3.8: Graph of cosine similarity and question answer pair (SSC)

As for the cosine graph (figure 3.8) of S.S.C scripts, it expresses a rather broader range of relevance than the BGS scripts as the score is assembled around 0.2-0.8. It has a wider range than the previous batch of creative QA.

### 3.3 Feature Engineering

#### 3.3.1 Criteria Problem

After analyzing the answer scripts we realized that,

- BNG103 datasets that we had collected were not annotated based on the rubric provided by the faculty, meaning, for each of the answers, marks were not provided based on the criteria, and for creative QA no rubric or annotation was provided. Rather, each answer was marked as a whole which is the usual occurrence. Due to this, we didn't have any explicit values for the rubric criterion which naturally makes it incredibly difficult for us to analyze our dataset.
- But as the creative QA scripts were manually digitized by us, we gained a general concept of the criteria.
- Based on our knowledge we decided to do feature extraction to better understand our dataset.

Since we did not obtain the exact rubric for grading the school scripts, we have created a rubric of our own which is based on the generalized system of the average school in Bangladesh and the observations given below:

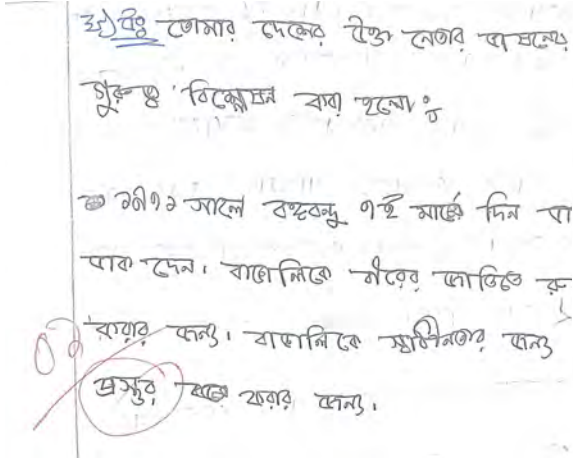


Figure 3.9: Spelling Criteria

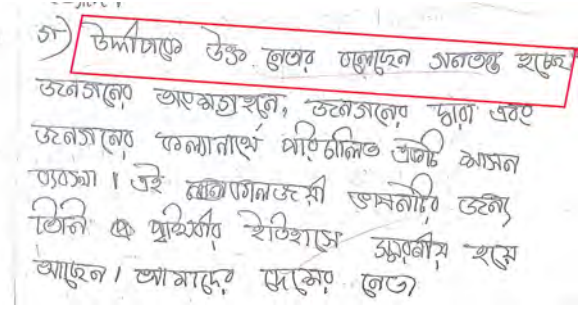


Figure 3.10: Grammar Criteria

### 3.3.2 Identification of Criteria

The quality of each of the essays was judged based on certain fixed criteria. The following rubric is how all the essays were marked:

Criteria	Marks
Grammar	3
Spelling	3
Knowledge about the topic or relevance of the answer with the question	4

Table 3.1: Creative QA Marks Distribution

Criteria	Marks
Introduction	1
Grammar	1
Spelling	1
Knowledge about the topic or relevance of the answer with the question	3
Own point of view	3
Conclusion	1

Table 3.2: BNG103 Marks Distribution Provided By Faculty

Due to the normalization of the 0-10 range, the mark distribution is done like above. For the BNG103 dataset, the quality of each of the essays was judged based on certain fixed criteria.

#### Introduction

After determining the criteria for BNG103 dataset, grading the answers on different features is the key difficulty. The pattern of BNG103 answers consists of an introduction with basic information about the topic or the poet or writer of the content on which the question is based. This information needs to be recognized for grading purposes.

In this case, to detect the presence of these entities, NER (Name Entity Recognition) was the reasonable solution.

Our expedition towards the improvement of automated essay scoring in Bengali led us to adopt and utilize BNL (Bengali CRF NER). BNL utilizes a rule-based Conditional Random Fields (CRF) approach. We attempted two methods which are BNL and Multilingual BERT but Multilingual BERT has an issue. For example, in figure 3.11, the word Rabindranath is identified and labeled as a person by BNL but Multilingual BERT was unable to do so which is why we decided to use the BNL method.

Answer	ner_tag_using_bnl	ner_tag_using_pretrained_mbert-bengali-ner_model
<p>কম চরিত্র ও পরিমিত ভাষার মাধ্যমে সম্পূর্ণ মনের ভাব করা যায় ছোটগল্পে। রবীন্দ্রনাথ ঠাকুর এবং ছোটগল্প দুটি ওতপ্রোতভাবে জড়িত। রবীন্দ্রনাথ ঠাকুর ১৮৯৪ সালের ২৭ জুন শিলাইদহ থেকে ইন্দ্রিা দেবীকে লেখা চিঠিতে ছোটগল্পের প্রতি নিজের ভালোবাসা, তার মনের শান্তি এবং নিজের একলা থাকার সঙ্গী হিসেবে আখ্যায়িত করেন। তিনি কবিতা, গান, প্রবন্ধ, নাটক প্রভৃতিতে সফলতার স্বাক্ষর রেখেছিলেন।</p>	<p>[('কম', 'O'), ('চরিত্র', 'O'), ('ও', 'O'), ('পরিমিত', 'O'), ('ভাষার', 'O'), ('মাধ্যমে', 'O'), ('সম্পূর্ণ', 'O'), ('মনের', 'O'), ('ভাব', 'O'), ('করা', 'O'), ('যায়', 'O'), ('ছোটগল্পে', 'O'), ('রবীন্দ্রনাথ', 'B-PER'), ('ঠাকুর', 'E-PER'), ('এবং', 'O'), ('ছোটগল্প', 'O'), ('দুটি', 'O'), ('ওতপ্রোতভাবে', 'O'), ('জড়িত', 'O'), ('রবীন্দ্রনাথ', 'B-PER'), ('ঠাকুর', 'E-PER'), ('১৮৯৪', 'O'), ('সালের', 'O'), ('২৭', 'O'), ('জুন', 'O'), ('শিলাইদহ', 'O'), ('থেকে', 'O'), ('ইন্দ্রিা', 'B-PER'), ('দেবীকে', 'E-PER'), ('লেখা', 'O'), ('চিঠিতে', 'O'), ('ছোটগল্পের', 'O'), ('প্রতি', 'O'), ('নিজের', 'O'), ('ভালোবাসা', 'O'), ('তার', 'O'), ('মনের', 'O'), ('শান্তি', 'O'), ('এবং', 'O'), ('নিজের', 'O'), ('একলা', 'O'), ('থাকার', 'O'), ('সঙ্গী', 'O'), ('হিসেবে', 'O'), ('আখ্যায়িত', 'O'), ('করেন', 'O'), ('তিনি', 'S-PER'), ('কবিতা', 'O'), ('গান', 'O'), ('প্রবন্ধ', 'O'), ('নাটক', 'O'), ('প্রভৃতিতে', 'O'), ('সফলতার', 'O'), ('স্বাক্ষর', 'O'), ('রেখেছিলেন', 'O')]</p>	<p>কম চরিত্র ও পরিমিত ভাষার মাধ্যমে সম্পূর্ণ মনের ভাব করা যায় ছোটগল্পে। রবীন্দ্রনাথ ঠাকুর এবং ছোটগল্প দুটি ওতপ্রোতভাবে জড়িত। রবীন্দ্রনাথ (LABEL_0) ##নাথ (LABEL_1) ঠাকুর ১৮৯৪ সালের ২৭ জুন (LABEL_0) শিলাইদহ (LABEL_5) থেকে (LABEL_0) ইন্দ্রিা (LABEL_1) দেবী (LABEL_2) ##কে লেখা চিঠিতে ছোটগল্পের প্রতি নিজের ভালোবাসা, তার মনের শান্তি এবং নিজের একলা থাকার সঙ্গী হিসেবে আখ্যায়িত করেন। তিনি কবিতা, গান, প্রবন্ধ, নাটক পরভৃতিতে সফলতার সবাঙ্কর রেখেছিলেন।</p>

Figure 3.11: NER tags

---

### Algorithm 1: Introduction Score

---

**Data:** answer

**Input :** answer

**Output:** score

```

1 Function extract_entities_and_get_scores(answer): begin
2   bn_ner ← BengaliNER();
3   bn_tokenizer ← BasicTokenizer();
4   tokens ← bn_tokenizer.tokenize(answer);
5   intro_text ← ' '.join(tokens[:50]);
6   ner_tags ← bn_ner.tag(intro_text);
7   if ner_tags then
8     has_per_loc ← any(entity[2] IN ['B-PER', 'E-PER', 'S-PER',
9     'B-LOC', 'E-LOC', 'S-LOC'] for entity IN ner_tags);
10  else
11    has_per_loc ← False;
12  score ← 1 if has_per_loc else 0;
13  return score;
14 score ← extract_entities_and_get_scores(answer);

```

---

After analyzing the introduction, we realized that the first 3 sentences generally

contain all the necessary information and knowledge required to score it. So, the above algorithm was used to score the introduction.

In terms of using the introduction algorithm, if we were able to detect a person or location, then the student obtains 1 mark. Otherwise, the student receives 0.

## Conclusion

After analyzing the dataset, we discovered that students write their own points of view in the conclusion paragraph which is why we carried out sentiment analysis on this particular segment. So, sentiment analysis is done but only on the last 3 sentences.

---

### Algorithm 2: Conclusion score

---

**Data:** *essay\_text*, *tokenizer*, *model*

**Input** : *essay\_text*, *tokenizer*, *model*

```
1 Function extract_last_3_sentences(essay_text): begin
2   | sentences ← [sentence.strip() for sentence in essay_text.replace('?',
   |   'daari').split("daari") if sentence.strip()];
3   | conclusion ← sentences[-3:];
4   | return conclusion;
5 Function SA_last_3_sentences(conclusion, tokenizer, model): begin
6   | sentiment_score ← probabilities[:, 1].item();
7   | sentiment_score ← "1.0" if sentiment_score > 0.5 else "0.0";
8   | binary_score ← 1.0 if sentiment_score > 0.5 else 0.0;
```

---

The *extract\_last\_3\_sentence* function here extracts the last 3 sentences from the answer. Characters such as "?", and "daari" are replaced with "daari" to separate the last 3 sentences from the answer and then return the sentence.

Then the part is tokenized and encoded with an input ID with a max length of 512 and truncated. Afterward, input ID is fed to the Bert model to obtain logits and then generate probabilities using softmax. The probabilities represent the sentiment score so as a result if it's greater than 0.5 then the marks of conclusion are set to 1 otherwise 0.

## Pos-Tagging

Essay or creative question answers are a combination of various information of a particular context and relevance to the scenario. To determine these factors, it is imperative to search and conclude all parts of speech in the content. Parts-of-Speech (POS) Tagging has been used for analyzing the part-of-speech distribution in the essay. For example, a high occurrence of nouns may indicate a focus on providing information, while adjectives and adverbs may contribute to expressing a point of view. So, Parts-of-Speech (POS) tagging is used in automated essay scoring for Bangla (or any language) to extract features related to the grammatical structure of the text.

Parts of Speech (POS) tagging allows us to gauge the quality of an essay in multiple ways. The grammatical and syntactic structure of sentences can be critically analyzed to determine how complex and correct they are in an essay. It can also evaluate how a writer adapts, based on the given situation, and applies grammatical rules in accordance. The writer's language proficiency can be further judged based on the vocabulary and language he uses to write an essay courtesy of the POS extraction features. Semantic roles can also be understood within sentences using POS tags. The relationship between nouns, verbs, adjectives, and other parts of speech can be evaluated which is quite major for the critical analysis of a writer's ability. In the same manner, if there are errors within the essay, POS tagging is able to pinpoint the errors made since those errors can significantly affect the quality of it. Combining POS features with specific rubrics, used to assess an essay, allows us to grade essays according to their relevance and with higher proficiency. Introducing POS tagging into AEG systems also increases feature diversity. The combination of all these factors is why POS tagging is so suitable for machine learning models that are designed to evaluate essays with a high level of accuracy.

Moreover, POS tagging provides information about the grammatical category of each word in a sentence.

ANSWER	POS TAGGING
<p>কম চরিত্র ও পরিমিত ভাষার মাধ্যমে সম্পূর্ণ মনের ভাব করা যায় ছোটগল্পে। রবীন্দ্রনাথ ঠাকুর এবং ছোটগল্প দুটি ওতপ্রোতভাবে জড়িত।</p>	<p>[('কম', 'JJ'), ('চরিত্র', 'NC'), ('ও', 'CCD'), (('পরিমিত', 'JJ'), ('ভাষার', 'NC'), ('মাধ্যমে', 'PP'), (('সম্পূর্ণ', 'JJ'), ('মনের', 'NC'), ('ভাব', 'NC'), (('করা', 'NV'), ('যায়', 'VM'), ('ছোটগল্পে', 'VM'), (('।', 'NP'), ('রবীন্দ্রনাথ', 'NP'), ('ঠাকুর', 'NP'), (('এবং', 'CCD'), ('ছোটগল্প', 'JJ'), ('দুটি', 'JQ'), (('ওতপ্রোতভাবে', 'AMN'), ('জড়িত', 'JJ'))]</p>

Figure 3.12: Example of POS Tagging

Here every word labeled with parts of speech where,

1. 'NC': Noun Common
2. 'JJ': Adjective
3. 'VM': Verb Main
4. 'CCD': Coordinating Conjunction
5. 'PP': Pronoun Personal
6. 'PRF': Pronoun Reflexive
7. 'JQ': Adjective Quantifier
8. 'AMN': Adverb of Manner
9. 'RDF': Adverb in Future

10. 'NP': Proper Noun

11. 'PU': Punctuation

## Grammar

The issue we faced in fulfilling the criteria of checking grammar was, there are no tools available for checking and assessing Bangla grammar. Meaning, there are no direct resources on correct tools. So, the next step of our feature extraction process is to find possible solutions.

**Translation of Bangla to English Sentences for Grammar Checking** - Our first approach was to employ Bangla to English Translation tools and use English grammar checkers to identify grammatical errors in the sentence. However, due to differences in grammatical structure and inaccuracy, the approach was swiftly changed.

Machine translation may not precisely capture the subtleties and linguistic nuances of Bengali sentences, leading to potential inaccuracies in the English translation. Furthermore, Bengali and English possess distinct grammatical structures, and translating sentences might result in altered structures that can pose challenges for accurate grammar analysis in English. The presence of context-specific or culturally specific language constructs in Bengali sentences might not be adequately preserved in translation, impacting the grammar checking accuracy. Moreover, the English grammar checker may generate false positives or false negatives as it is tailored for English grammar, and translated sentences may not conform perfectly to English grammatical rules.

Google Translate API and the LanguageTool library were used to translate and subsequently check grammar errors. The code defines a function named `translate_and_check_grammar` that takes a list of Bengali sentences as input. It then iterates through each Bengali sentence, translates it into English using Google Translate, and checks the grammar of the translated English sentence using LanguageTool.

```
Original Bengali Sentence,English Translation,Grammar Errors
আমি বাংলা ভাষা ভালোবাসি,I love Bangla language.,1. Rule ID: MORFOLOGIK_RULE_EN_US, Message: Possible spelling mistake found., Correction: ["Bangla"]
বাংলা সাক্ষাত সুন্দর।,Bengali literature is beautiful.,No Grammar Errors Found.
উদ্দেশ্যকে উল্লেখিত মাধ্যম ছাড়াও সামাজিকীকরণ আর মাধ্যম মানুষের জীবনে।,In addition to the medium mentioned in the stimulus, socialization and medium are in the life of the people.,No Grammar Errors Found.
আমি পরীক্ষা দিই ভয় পাই,I am afraid to take the test.,No Grammar Errors Found.
```

Figure 3.13: Result of Grammar check using translation

As you can see, the English translation of 1st sentence shows a grammatical error but it is a correct sentence, and 3rd sentence was translated incorrectly so the grammar error was not found even though it is grammatically incorrect.

**Grammar Checking Tool** - We created a grammatical checker for Bengali sentences using a trigram-based n-gram model and POS Tagging. The training phase involves utilizing a corpus built through web crawling on various Bengali websites,



comprising 10,000 sentences.

---

**Algorithm 3:** Grammar Score

---

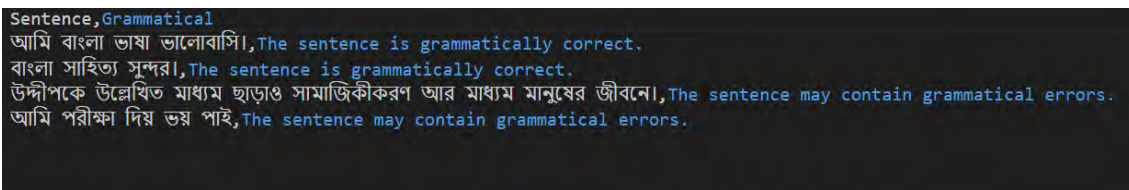
**Data:** essay

**Result:** Marks

```
1 sentences ← split_sentences(essay) + split_sentences(essay, '?');
2 total_sentences ← len(sentences);
3 wrong_grammar_sentences ← 0;
4 foreach sentence in sentences do
5   is_grammatical ← check_grammar(sentence, ngram_model, threshold);
6   if not is_grammatical then
7     wrong_grammar_sentences ← wrong_grammar_sentences + 1;
8 if total_sentences > 0 then
9   ratio_wrong_grammar ←  $\frac{\text{wrong\_grammar\_sentences}}{\text{total\_sentences}}$ ;
10  marks ← 1 - ratio_wrong_grammar;
11  OUTPUT("Marks: " + marks formatted with one decimal);
12 else
13  OUTPUT("No sentences found in the essay.");
```

---

Here, the answer is split according to space and "?" and the length of the sentence is calculated. Then the wrong\_grammar\_sentences is set to 0 and the split sentences each go through a for loop. In the loop, every sentence's grammar is checked in check\_grammar using the n-gram model and pos-tagging. If incorrect grammar is detected, the value of wrong\_grammar\_sentences is incremented to 1. Afterward, if the total\_sentences is greater than 0, then the algorithm finds the ratio of wrong grammar by dividing the wrong\_grammar\_sentences by total\_sentences and decides the score by excluding the ratio from full marks (1). For the creative QA, the full mark would be 3. If there is a mark then it will show in the output, otherwise, it gives "No sentences found in the essay."



The screenshot shows a dark background with light-colored text. It displays four lines of Bengali text, each followed by its grammatical status in English. The first two lines are marked as 'The sentence is grammatically correct.', while the last two are marked as 'The sentence may contain grammatical errors.' The Bengali text includes phrases like 'আমি বাংলা ভাষা ভালোবাসি।' and 'উদ্দীপকে উল্লেখিত মাধ্যম ছাড়াও সামাজিকীকরণ আর মাধ্যম মানুষের জীবনো.'

Figure 3.14: Working Grammar Tool

It is visible through the figure that all the grammatical errors were found in this approach.

### Spelling Check

The spell check in this essay grading system is developed by implementing the Bengali Word2Vec Model and BertForMaskedLM Model by Sagor Sarker. Here, spelling mistakes are detected using BertForMaskedLM by Sagor Sarker.

---

**Algorithm 4: Detect Misspelled Words and Calculate Spelling Marks**

---

**Data:** row  
**Input :** row

```
1 Function detect_misspelled(row):  
2 essay ← row["essay"];  
3 tokenized_text ← basic_tokenizer.tokenize(essay);  
4 misspelled_words ← [];  
5 for word in tokenized_text do  
6   | vector ← bwv.get_word_vector(word);  
7   | KeyError misspelled_words.append(word);  
8 total_words ← len(tokenized_text);  
9 misspelled_percentage ←  $\frac{\text{len}(\text{misspelled\_words})}{\text{total\_words}}$ ;  
10 rounded_misspelled_percentage ← round(misspelled_percentage, 1);  
11 spelling_marks ← 1 – rounded_misspelled_percentage;  
12 return spelling_marks;
```

---

The detect\_misspelled function works by calculating the length of tokenized text which is considered the total\_words the length of misspelled\_words is divided by total\_words to count the percentage of misspellings and is rounded to 1. Then the percentage is deducted from the full marks (1) to grade according to the criteria and that is for essay scripts. For school scripts, the misspell percentage will be deducted from 3.

## Grade on Contextual Knowledge and Relevance

**Word-Embedding** - The contextual meaning of the content is necessary to know to grade essays but as there are accurate tools available, we decided to work by understanding the meaning of words for word embeddings as our approach.

Word embeddings are used to capture contextual embeddings for each word in the cleaned answers and questions of an essay. According to the context of the full text, these embeddings provide semantic information about the words. Moreover, for automated essay scoring, these embeddings could potentially be used as features for a machine-learning model. The criterion used to judge the quality of the essays is identified as corresponding patterns in the embeddings by the model. The representation of words is more characterized due to the embeddings. It considers the context within each essay which is extremely useful for understanding the overall meaning and quality of the text.

Bangla BERT model creates embedding of both question and answer and afterwards, the model's tokenizer is utilized to tokenize and send it to the model.

**Cosine Similarity** - The calculation of cosine determines the relativity between the question and answer. The embedding vector's angle is calculated and shows a quantitative representation of the relevance.

The combination of these two methods defines our approach to scoring the answer

based on contextual knowledge and similarity with the given question and textbook knowledge. The code is given below:

---

**Algorithm 5:** Compute Context Knowledge and Relevance Score

---

**Data:** *df*, *tokenizer\_bangla*, *model\_bangla*

**Result:** *relevance\_score*

```
1 essay_embeddings_bangla, question_embeddings_bangla ←  
   get_bangla_bert_embeddings(df['cleaned_answer']),  
2 get_bangla_bert_embeddings(df['Question']);  
   cosine_similarities_bangla ←  
   cosine_similarity(essay_embeddings_bangla,  
   question_embeddings_bangla);  
3 relevance_score ← np.mean(cosine_similarities_bangla, axis = 1);  
4 if relevance_score > 0.8 then  
5   | relevance_score ← 3;  
6 else if 0.5 < relevance_score ≤ 0.8 then  
7   | relevance_score ← 2;  
8 else  
9   | relevance_score ← 0;
```

---

As the algorithm shows, *cleaned\_answer* and *Question* are given to the BERT model and output embeddings which hold vectors with values that put meaning to the content. Both answers and questions are embedded and then the angle between these two embedding vectors is calculated to get the relevance score using *cosine\_similarity* function. If the relevance score is greater than 0.8 then it is graded 3 and if it's greater than 0.5 and less than or equal to 0.8 then it is graded 2, otherwise, it is set to 0. Contextual knowledge and relevance are used in both essay and creative question so it is normalized to 0-1 for creative QA.

### Point of View

This section allows students to express their opinions using information and critical reasoning. Generating their own point of view refers to emotions and sentiments which is why sentiment analysis seemed a fair approach for scoring this criteria. As a result the algorithm 6 algorithm was used. The feature's grading process is similar to conclusion. The difference lies in extraction of the middle part which is done by excluding the first and last 3 sentences, meaning the introduction and conclusion.

---

**Algorithm 6: Own Point of View Analysis**

---

**Data:** *essay\_text*  
**Input :** *essay\_text*  
**Output:** *score*

```
1 Function extract_middle_sentences(essay_text): begin
2   sentences  $\leftarrow$  [sentence.strip() FOR sentence IN essay_text.replace('?',
   'daari').split("daari") IF sentence.strip()];
3   if len(sentences)  $\leq$  4 then
4     OUTPUT("The essay is too short to extract middle sentences.");
5     RETURN [];
6   middle_sentences  $\leftarrow$  sentences[1:-3];
7   RETURN middle_sentences;

8 Function analyze_sentiment_middle_sentences(middle_sentences):
   begin
9   FOR sentence IN middle_sentences: OUTPUT(sentence);
10  INPUT_ids  $\leftarrow$  tokenizer.encode(" ".join(middle_sentences),
   RETURN_tensors="pt", max_length=512, truncation=True);
11  with torch.no_grad(): outputs  $\leftarrow$  model(INPUT_ids);
12  logits  $\leftarrow$  outputs.logits;
13  probabilities  $\leftarrow$  torch.softmax(logits, dim=1);
14  sentiment_score  $\leftarrow$  probabilities[:, 1].item();
15  IF sentiment_score > 0.7: score  $\leftarrow$  3.0;
16  ELSEIF sentiment_score > 0.5: score  $\leftarrow$  2.0;
17  ELSE: score  $\leftarrow$  0.0;

18 score  $\leftarrow$ 
   analyze_sentiment_middle_sentences(extract_middle_sentences(essay_text));
```

---

Before excluding the introduction and conclusion, the algorithm checks if there are fewer than 4 sentences. If yes, then it returns an empty array otherwise the middle part is extracted. This criterion is only required for essay scripts. That extracted part is sent for sentiment analysis. If the sentiment score is above 0.7, then the score is set to 3 which is the full mark. Also, if the score is less than 0.7 but greater than 0.5, then 2 is assigned grade. Otherwise, the grade is 0.

## 3.4 Model

### 3.4.1 Why BERT Model?

BERT is a state-of-the-art transformer-based model designed for comprehending tasks on natural language processing. The crucial point for tasks such as essay scoring is that contextual information can be captured considering both left and right context in a sentence which BERT is capable of. The bidirectional attention of BERT allows the entire context of a word to be considered when generating word representations. This is essential for understanding the intricate relationships between words in an essay. BERT can also analyze the coherence and flow of an essay courtesy of its capability to capture long-range dependencies in a sentence.

### 3.4.2 Why not other models?

Analyzing the distinctive relationship between words might be a struggle for traditional models such as TF-IDF, Naive Bayes, or even simpler neural network architectures. Often, these models handle words independently which is a major limitation for tasks where contextual understanding is critical. GLSA, for example, relies typically on a bag-of-words representation and may not consider the sequential relationships between words so it may struggle in contextual understanding at the same standard as BERT. Models such as bag-of-words or simpler neural networks may fail in the effective capture of semantic relationships. Instead, they focus on local context or individual sentences ultimately missing the broader picture. Other models lack the benefits of transfer learning which is quite resourceful when working with data whose annotation is either limited or none. General language patterns can be learned by models pre-trained on a large dataset before they are fine-tuned for a task-specific dataset.

### 3.4.3 Why Pre-trained Bangla Models?

Pre-trained Bangla models have already learned language patterns, intricacies, and context relevant to Bangla since they are specifically trained on Bangla language data making them an obvious choice for tasks involving Bangla text. Since pre-trained models capture the linguistic details of a language better than models trained on general-purpose corpora, the performance of language-specific tasks is often improved. These pre-trained models are more data efficient for specific tasks in Bangla compared to general language corpora-based models. There is less requirement for fine-tuning and better comprehension of the language due to these models.

### 3.4.4 Pre-trained BERT Model

BERT is a framework, often used in machine learning that focuses on NLP. It can subsequently be fine-tuned using question-and-answer datasets. We should consider incorporating BERT into our topic because it has similarities with projects that utilize the BERT model for automated essay scoring. It was designed for long input context and pre-trained with multi-task objectives such as sentiment analysis, and text classification [18].

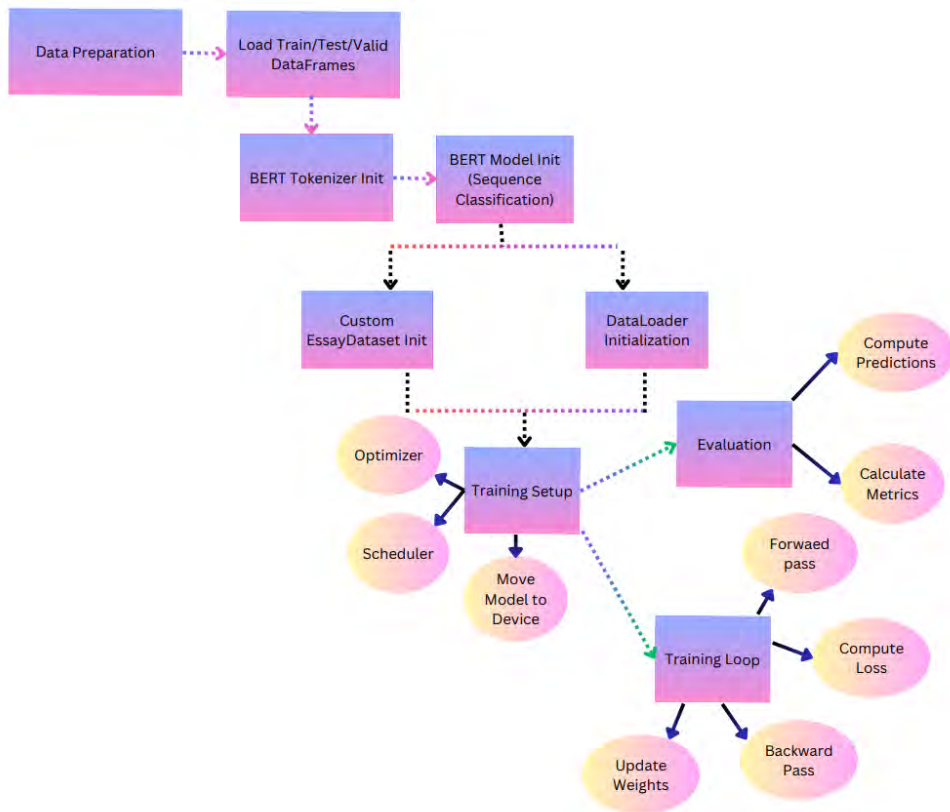


Figure 3.15: BERT Model

The architecture consists of an encoder with multiple layers of self-attention mechanisms, enabling it to capture contextual information bidirectionally. This bidirectional context understanding is crucial for understanding the meaning of the words used in language. The BERT model is the base of the new model developed by Sagor Sarker which is called “BanglaBertBase” where the model is trained by Bengali common crawl corpus from OSCAR and Bengali Wikipedia Dumb dataset. The model follows the basic architecture of the BERT Model so to exclude the known factors, readers can refer to [13].

At first, the combined dataset containing cleaned essay or answer text and marks is loaded. The text data is then tokenized using the BERT tokenizer with a specified sequence length and split into training and validation sets. As we can see in the diagram, there is a class “Custom Dataset Class” where the question and cleaned answer are tokenized as ”input\_ids” and ”attention mask”. Also, marks are marked as ”labels”. After that, data loaders for batch processing are created, and the BERT model is initialized for regression. Then, the optimizer and learning rate scheduler are defined, and the model will run the loop of training for specific epochs (20). Lastly, the model is evaluated on the validation set using RMSE and  $R^2$  scores.

1

<sup>1</sup><https://huggingface.co/sagorsarker/bangla-bert-base>

### 3.4.5 Pre-trained BERT Model (Fine-tuned)

There are a few points to observe that,

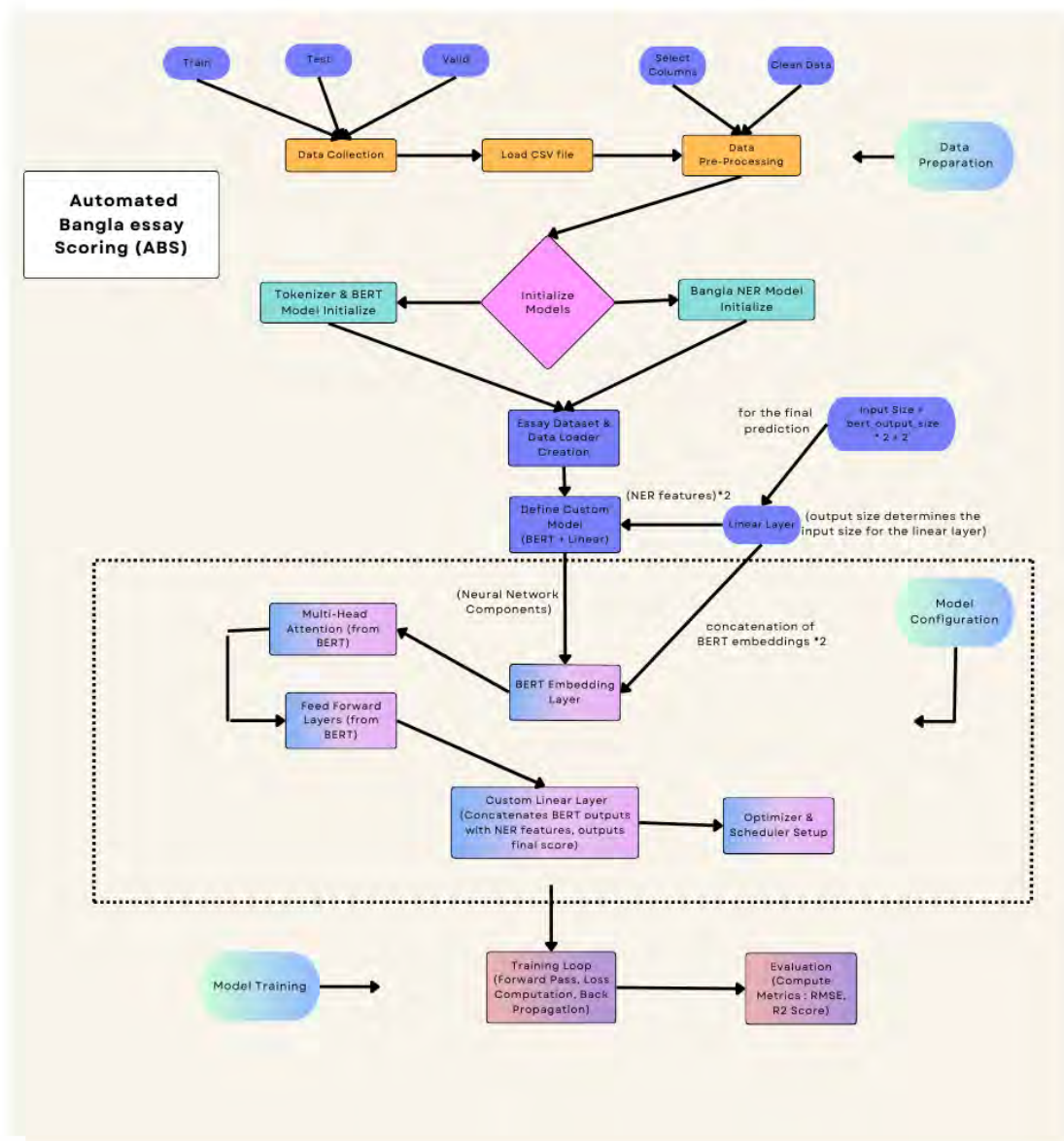


Figure 3.16: Fine-Tuned BERT Model

- The BERT model was designed to perform various tasks of NLP, meaning, with added layers and pre-training, a basic architectural model can function for particular tasks.
- Both types of datasets in our research should be scored considering different criteria of the institutions, as well as the general rule of marks distribution in the Bangladesh Education system.
- As we have mentioned criteria such as introduction, and point of view, we would need to utilize various other tools to fine-tune the model.

So, to strengthen the transfer learning of the model, the Bangla BERT Model provided by Sagor Sarker was fine-tuned.

As you can see before, the fine-tuned model goes from the data preparation to the tokenizer and BERT model initialization as before. In addition to the BERT model and tokenizer, the Bangla NER model is initialized from the BNLP library. The structure of the Custom Model class changes due to integrating the NER feature linear layer. The linear layer provides NER feature integration and BERT embeddings for which its input size is `bert_output_size*2` which is the concatenation 2 embeddings from the model and the 2 NER features. Here, the NER features identify people and locations. It produces the output for both question and answer using NER features. It is done to improve the model's ability to better understand the content and also to predict the marks with greater accuracy.

In the neural network, the output of the embedding layer goes through Multi-Head attention to learn the meaning of words in an extensive range. Afterward, the feed forward layer changes the attention vector into a form that can be understood by the next encoder and the custom linear layer outputs the score. The training loop continues with the optimizer and scheduler controlling the learning rate and the evaluation is done using RMSE and  $R^2$ .

### 3.4.6 Pre-trained ALBERT Model

This model is implemented using a similar architecture to the BERT Model, where two techniques are applied to reduce limitations which are factorized embedding parameterization and crosslayer parameter sharing. Now the reasons to utilize this model were,

- Implementation of the BERT model was a significant improvement.
- A standard Bangla AEG system should be able to handle a high amount of datasets and as ALBERT was a part of the BERT model to reduce the memory limitations, the model was chosen as an option.

We have implemented the model and it didn't bring satisfactory results pointing out that, the model lacks the capacity to capture certain linguistic nuances that are present in our Bangla dataset.



# Chapter 4

## Result Analysis

### 4.1 Model Result

Table 4.1: Essay Scoring Evaluation

Essay	BERT		Fine-tuned BERT		ALBERT	
	RMSE	R <sup>2</sup>	RMSE	R <sup>2</sup>	RMSE	R <sup>2</sup>
Test	0.6076	0.8295	0.6056	0.8216	1.7269	-0.4550
Validation	0.6136	0.0010	0.6078	0.7016	1.9148	-7.3802

Table 4.2: Creative QA Score Evaluation (Test)

Test Metrics	Class-8 (B.G.S)		Class-9 (B.G.S)		S.S.C	
	RMSE	R <sup>2</sup>	RMSE	R <sup>2</sup>	RMSE	R <sup>2</sup>
BERT	0.6109	0.8211	0.6398	0.8219	0.6024	0.8351
BERT (Fine-tuned)	0.5234	0.8213	0.6411	0.8236	0.6155	0.6783
Albert	1.8022	-0.5521	1.2399	-0.6810	1.9418	-0.5937

Table 4.3: Creative QA Score Evaluation (Validation)

Validation Metrics	Class-8 (B.G.S)		Class-9 (B.G.S)		S.S.C	
	RMSE	R <sup>2</sup>	RMSE	R <sup>2</sup>	RMSE	R <sup>2</sup>
BERT	0.6040	0.7211	0.6278	0.8235	0.6794	0.3475
BERT (Fine-tuned)	0.5164	0.6981	0.6449	0.8086	0.6255	0.1337
Albert	1.7026	-0.4321	1.3982	-0.6611	1.8843	-0.0025

We used two scores, RMSE and R<sup>2</sup> to assess how well the model performs. The R<sup>2</sup> value indicates the extent to which our model can account for the variation observed in real grades where RSS is the sum of squares of residuals and TSS is the total sum of squares.

$$R^2 = 1 - \frac{RSS}{TSS} \quad (4.1)$$

As shown in Tables 4.1,4.2 and 4.2, the fine-tuned BERT model's RMSE value for both test set and validation set metrics is greater than the SagorSarkar/Bangla Bert-base model, indicating a higher RMSE score. The  $R^2$  value for the validation set metrics was better for the fine-tuned model. However, the test set did not perform as well as the validation set, as the test set may have had a larger degree of unpredictability than the validation set after splitting. This higher variability in the test dataset might have introduced additional challenges for the model, making it more difficult to generalize accurately to unseen data. In addition, the scores of the Albert model go beyond the proper accuracy score, making it unfit for our AEG model. However, for classes 8 & 9 the RMSE and  $R^2$  value shows better results for the fine-tuned model. In the case of SSC Bangla 1st paper, the RMSE value is slightly higher &  $R^2$  value is slightly lower for our fine-tuned model which could be due to the fact that the dataset may include some instances of a high degree of variability, or complex patterns that are challenging for the model to capture accurately.

## 4.2 Comparison among Previous Researches

1. **Dataset:** The work of [16] was done based on 20 questions where 10 students answered only 2 questions out of 20. So, according to this the training and test of the system was done using only 20 answers of 300-350 words. There is also research conducted by Hussain et al. in [8] where 150 scripts of “Bangladesher Shadhinota Songram” and 200 scripts of “Karigori Shiksha” are used for training and evaluation. The same dataset was used in [7] but quantity differed by 30 and 20 scripts respectively.

Whereas, our work was done on 324 opinion-based essays and around 250 scripts of creative question answers. The point to note here is every script has 7 creative questions and as we have conducted research on the 4th question, every script has 7 answers on average.

- So, technically the number comes down to around 1750 samples consisting of 150-200 words. So, the quantity is far larger compared to other works done on the Bangla essay scoring system.
  - There is also the fact that the research in [7] used synthetic samples to test the system and this approach was not taken in our research to maintain the standard of the work.
  - We have operated on creative question answers which haven’t been done in the previous research. So it can provide a new perspective on working with these types of content.
  - All the scripts employed in this paper are manually graded by experts in their fields, so the authenticity is visible.
2. **Evaluation criteria:** Previous research defined criteria through spelling check, grammar check, sentence structure, and minimum relevance with existing open domain[8] [7].In[16], keyword comparison using both open and closed corpus dominated the grading system. However, in this paper, new criteria have been evaluated using accessible NLP tools. It was focused mainly on the touch of humanity in the essay grading system.

- Introduction, knowledge about the topic, point of view, and conclusion were introduced as criteria for Bangla essays.
  - And, for creative QA relevance with the given scenario was an integral part as the system needs to be trained on specific topics for scoring the answers.
3. **Approach:** In [16], finding keywords was essential to grade essays by using word frequency whereas in other works SVD and n-gram were used to predict the grammar check and understand the context of word basis [8] [7]. Our work was centered around technical issues rather than mechanical ones such as spelling, and grammar. With given criteria and different datasets, it is difficult to choose from existing approaches. For example, point of view criteria refers to expressing opinion utilizing textbook context. It is highly encouraged by experts because It means the student has understood the content well

enough to use it to reason their point of view. So, sentiment analysis was used as a concept to determine whether they had expressed their point or not. According to experts, an introduction to an essay should contain information regarding the topic and to identify it, NER was used. It helps in locating the entity and determining if knowledge of the topic was present or not. Different criteria determined different approaches.

### 4.3 Limitation

1. The current word corpus for spelling contains the correct spelling of Bangla words from web browsers, which is why it can't detect correct words according to the Bangladesh School Education syllabus.
2. The existing tokenization tools are not sufficient enough to accurately score Bangla Essay and Creative question answers.
3. Although our dataset provides authenticity and actuality, its amount is far less to build a workable AEG system.
4. The lack of professional help from teachers caused ambiguity in mark distribution and automation of scoring.
5. Inadequate amount of manpower for digitization was time-consuming, causing delays in producing results.

# Chapter 5

## Future Work

The future goal remains to collect more real-time essays and creative question-answer scripts in order to develop the Bangla Essay Scoring System corpus to develop a deployable AEG system for Bangla. The result would be better with the help of the Schools and the Education Ministry of Bangladesh to collect quality datasets. We also plan to develop the Bangla Bert Model as fine-tuning is done considering individual tasks So, to get a working scoring system for the specified dataset, more work is needed in fine-tuning the current Transformation model. With these motivations, we plan to bring about an industry standard Bangla Bert Model to develop the AEG system for Bangla.

# Chapter 6

## Conclusion

Automated Essay Grading (AEG) in Bangla Language is currently in the early stages of development. Due to the lack of study done in this area, particularly for Bangla essay checkers, neither adequate tools for recognizing Bangla grammar nor a tool that will comprehend the level of the writers are available. After an in-depth assessment of the available research articles on AEG and implying the knowledge of a versatile dataset, we have stumbled upon some constraints. So, we have attempted to address these constraints through an in-depth analysis of requirements for grading Bangla Essays, increasing the collection of datasets by collecting both easy and creative question-based content, and contributing to the existing work by trying different outlooks to make significant developments in the current AEG system. As a result, we were able to contribute to designing a competent system for Bangla Essay Grading to make the task of a teacher much easier and provide better checking quality. Therefore, our initiative is to create a developed Bangla Essay Grading System and collaborate with the education board and private organizations to boost our research work.

# Bibliography

- [1] J. Burstein, K. Kukich, S. Wolff, *et al.*, “Automated scoring using a hybrid feature identification technique,” in *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, 1998, pp. 206–210.
- [2] J. Burstein, K. Kukich, S. Wolff, C. Lu, and M. Chodorow, “Enriching automated essay scoring using discourse marking.,” 2001.
- [3] J. Burstein, M. Chodorow, and C. Leacock, “Automated essay evaluation: The criterion online writing service,” *AI Magazine*, vol. 25, no. 3, p. 27, Sep. 2004. DOI: 10.1609/aimag.v25i3.1774. [Online]. Available: <https://ojs.aaai.org/index.php/aimagazine/article/view/1774>.
- [4] T. Kakkonen, N. Myller, J. Timonen, and E. Sutinen, “Automatic essay grading with probabilistic latent semantic analysis,” in *Proceedings of the second workshop on Building Educational Applications Using NLP*, 2005, pp. 29–36.
- [5] Y. Attali and J. Burstein, “Automated essay scoring with e-rater® v.2,” *The Journal of Technology, Learning and Assessment*, vol. 4, no. 3, Feb. 2006. [Online]. Available: <https://ejournals.bc.edu/index.php/jtla/article/view/1650>.
- [6] S. Ghosh, “Design of an automated essay grading (aeg) system in indian context,” Dec. 2008, pp. 1–6. DOI: 10.1109/TENCON.2008.4766677.
- [7] M. Islam and A. Latiful Haque, “Automated essay scoring using generalized latent semantic analysis,” *Journal of Computers*, vol. 7, Mar. 2012. DOI: 10.4304/jcp.7.3.616-626.
- [8] M. M. Islam and A. S. M. L. Hoque, “Automated bangla essay scoring system: Abess,” in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, 2013, pp. 1–5. DOI: 10.1109/ICIEV.2013.6572694.
- [9] R. Östling, A. Smolentzov, B. Tyrefors Hinnerich, and E. Höglin, “Automated essay scoring for Swedish,” in *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 42–47. [Online]. Available: <https://aclanthology.org/W13-1705>.
- [10] 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*, 2016, pp. 1882–1891.
- [11] K. Zupanc and Z. Bosnic, “Advances in the field of automated essay evaluation,” *Informatica*, vol. 39, no. 4, 2016.



- [12] M. Cozma, A. M. Butnaru, and R. T. Ionescu, “Automated essay scoring with string kernels and word embeddings,” *arXiv preprint arXiv:1804.07954*, 2018.
- [13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [14] S. Khatun and M. Hoque, “Semantic analysis of bengali sentences,” Sep. 2018, pp. 1–6. DOI: 10.1109/ICBSLP.2018.8554726.
- [15] V. Ramalingam, A. Pandian, P. Chetry, and H. Nigam, “Automated essay grading using machine learning algorithm,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1000, 2018, p. 012 030.
- [16] M. G. Hussain, S. Kabir, T. A. Mahmud, A. Khatun, and M. J. Islam, “Assessment of bangla descriptive answer script digitally,” in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2019, pp. 1–4. DOI: 10.1109/ICBSLP47725.2019.202042.
- [17] I. Ashrafi, M. Mohammad, A. Shawkat, *et al.*, “Banner: A cost-sensitive contextualized model for bangla named entity recognition,” *IEEE Access*, vol. PP, pp. 1–1, Mar. 2020. DOI: 10.1109/ACCESS.2020.2982427.
- [18] A. M. S. Tasmiah Tahsin Mayeesha and R. M. Rahman, “Deep learning based question answering system in bengali,” *Journal of Information and Telecommunication*, vol. 5, no. 2, pp. 145–178, 2021. DOI: 10.1080/24751839.2020.1833136.
- [19] N. R. Bhowmik, M. Arifuzzaman, and M. R. H. Mondal, “Sentiment analysis on bangla text using extended lexicon dictionary and deep learning algorithms,” *Array*, vol. 13, p. 100 123, 2022, ISSN: 2590-0056. DOI: <https://doi.org/10.1016/j.array.2021.100123>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S259000562100059X>.
- [20] Y.-J. Jong, Y.-J. Kim, and O.-C. Ri, “Improving performance of automated essay scoring by using back-translation essays and adjusted scores,” *Mathematical Problems in Engineering*, vol. 2022, 2022.
- [21] M. Kowsher, A. A. Sami, N. J. Prottasha, M. S. Arefin, P. K. Dhar, and T. Koshiba, “Bangla-bert: Transformer-based efficient model for transfer learning and language understanding,” *IEEE Access*, vol. 10, pp. 91 855–91 870, 2022. DOI: 10.1109/ACCESS.2022.3197662.