applied sciences

MDPI

Article

Combining Balancing Dataset and SentenceTransformers to Improve Short Answer Grading Performance

Maresha Caroline Wijanto * and Hwan-Seung Yong

Department of Artificial Intelligence and Software, Ewha Womans University, Seoul 03760, Republic of Korea; hsyong@ewha.ac.kr

* Correspondence: mareshacw@ewhain.net

Abstract: Short-answer questions can encourage students to express their understanding. However, these answers can vary widely, leading to subjective assessments. Automatic short answer grading (ASAG) has become an important field of research. Recent studies have demonstrated a good performance using computationally expensive models. Additionally, available datasets are often unbalanced in terms of quantity. This research attempts to combine a simpler SentenceTransformers model with a balanced dataset, using prompt engineering in GPT to generate new sentences. Our recommended model also tries to fine-tune several hyperparameters to achieve optimal results. The research results show that the relatively small-sized all-distilroberta-v1 model can achieve a Pearson correlation value of 0.9586. The RMSE, F1-score, and accuracy score also provide better performances. This model is combined with the fine-tuning of hyperparameters, such as the use of gradient checkpointing, the split-size ratio for testing and training data, and the pre-processing steps. The best result is obtained when the new generated dataset from the GPT data augmentation is implemented. The newly generated dataset from GPT data augmentation achieves a cosine similarity score of 0.8 for the correct category. When applied to other datasets, our proposed method also shows an improved performance. Therefore, we conclude that a relatively small-sized model combined with the fine-tuning of the appropriate hyperparameters and a balanced dataset can provide performance results that surpass other models that require larger resources and are computationally expensive.

Keywords: short answer question; automatic short answer grading (ASAG); SentenceTransformers; dataset balancing; GPT; fine-tuning

check for updates

Citation: Wijanto, M.C.; Yong, H.-S. Combining Balancing Dataset and SentenceTransformers to Improve Short Answer Grading Performance. *Appl. Sci.* **2024**, *14*, 4532. https://doi.org/10.3390/app14114532

Academic Editor: Andrea Prati

Received: 23 April 2024
Revised: 18 May 2024
Accepted: 23 May 2024
Published: 25 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As a result of the COVID-19 pandemic that emerged at the end of 2019, E-learning or web-based distance learning platforms have become a viable alternative to facilitate the learning process [1]. Within this learning process, knowledge assessment plays a pivotal role in ensuring effective teaching [2]. Open-ended questions have been identified as a valuable method for determining students' level of knowledge and encouraging them to express their thoughts, perspectives, and experiences in their own words [1]. By inviting open-ended responses, teachers gain a more accurate and comprehensive insight into how students grasp domain-specific knowledge [3]. However, manually scoring these responses can introduce inconsistencies, as scoring may vary among markers or from one student to another [2]. Additionally, expecting a single definitive response to an open-ended question proves challenging for teachers, due to variations in students' vocabulary and writing structures [4]. This can lead to subjective judgements about the answers and compromise the objectivity of the assessment process [5].

According to Burrows et al. in Ref. [6], short answers have the following characteristics: the answer should not inferred just from the question's words (requiring external knowledge); the answer should be given in natural language; the length of the answer typically spans from one phrase to one paragraph; the content of the answer is relevant

to the subject domain; and the answer is structured as closed-ended yet is not rigidly defined. However, some short-answer questions require students to express their subjective viewpoints within a defined context. Hence, short-answer questions also refer to semi-open-ended questions [7].

The grading system for short answers poses inherent challenges compared to automated multiple-choice grading systems. It is essential to thoroughly examine the nuances and variations in these answers to ensure accurate assessment [8]. The advancements in natural language processing (NLP) and machine learning applications have spurred interest among educators in creating exams comprising open-ended questions that can be automatically evaluated for a large number of students [5].

Automatic short answer grading (ASAG) is an emerging field of research, reflecting the educational sector's increasing adoption of technology to aid students and professionals. ASAG systems hold potential as valuable resources for educators, facilitating the enhanced integration of open-ended questions and providing more objective assessments for both formative and summative evaluations [9]. ASAG functions by analyzing students' answers in relation to a given question and the desired answer, as illustrated in Figure 1.

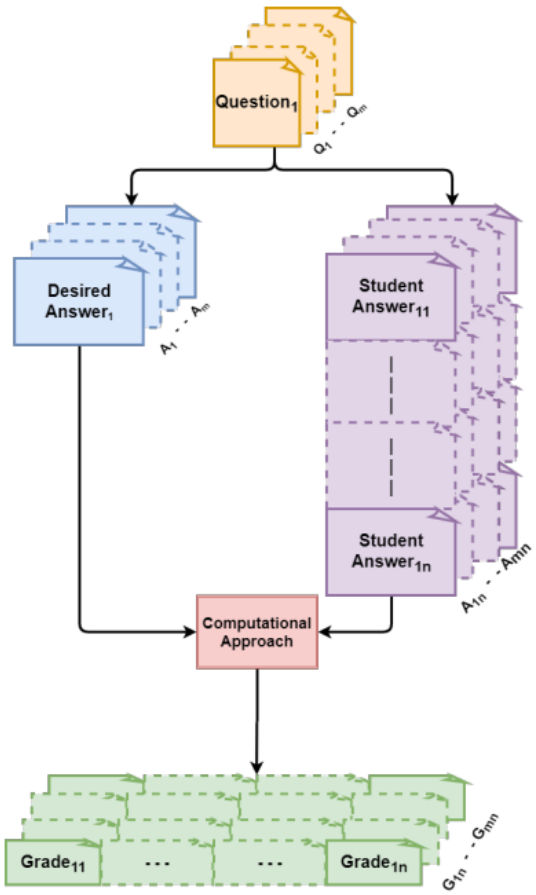


Figure 1. ASAG pipeline [10].

The recent advancements in natural language processing (NLP) and deep learning have introduced promising methodologies and frameworks capable of addressing various tasks. Across numerous NLP tasks, including ASAG, language models (LMs) have demonstrated considerable success. In modern approaches, LMs are trained using neural networks. Initial neural models were based on recurrent neural networks (RNNs), like long short-term memory networks (LSTMs and BiLSTm) [3]. The development of large language models like BERT (Bidirectional Encoder Representations from Transformers), based on the transformer architecture, and the increasing adoption of transfer learning have been instrumental in constructing custom ASAG systems [11].

Transformer employs self-attention for natural language processing, enabling the parallel computation of input and output vectors and addressing the sequential processing limitations of recurrent neural network (RNN), convolutional neural network (CNN), and long short-term memory (LSTM) approaches [12]. However, this self-attention mechanism can be computationally expensive. In the literature, ASAG studies often measure success by high correlations and a minimal loss value on standard benchmark tests using widely accessible datasets [5]. Many researchers indicate that the performance of ASAG systems is closely tied to the volume of training data available [13].

Reimers et al. [14], argued that, while the BERT and RoBERTa language models have a new state-of-the-art performance in sentence-pair regression tasks like semantic textual similarity, enabling the input of all the sentences into the network, the computational overhead is considerable. Though ASAG is crucial, implementing these expensive models may pose challenges. In this research, we explore some other Sentence-Bidirectional Encoder Representations (SBERT) models as mentioned in [15], and then propose a simpler model by fine-tuning certain hyperparameters to optimize the ASAG performance.

The available datasets exhibit an imbalance in the distribution of data across different labels. For instance, The SciEnts Bank (SEB) dataset [16] has a correct answer ratio of 39.9%, with label 4 representing the correct answer, while labels 0–3 are considered incorrect answers. And in contrast, the Mohler dataset from the University of North Texas predominantly contains correct answers, with approximately 78% coming from labels 4–5 only. As mentioned in Ref. [17], achieving a balanced dataset is crucial for developing optimal models, although this is challenging in practice. Augmentation, a method within oversampling, involves enhancing an existing dataset by adding supplementary data. Various methods used for augmenting data in NLP include random deletion, synonym replacement, random swap, and back translation [13,17]. In this research, we augment the dataset by utilizing GPT to paraphrase the answer, serving as an additional strategy of synonyms replacement. Despite GPT's inability to directly augment data, prompts can be utilized to generate synonyms and antonyms of words.

This paper is organized as follows: Section 2 reviews all works related to ASAG. Section 3 presents the proposed methods, including the datasets, evaluation metrics, and experiment setup used in this research. Section 4 presents a proof-of-concept implementation of the system and discussions of the experimental results. Section 5 concludes with all of the achievements from these experiments.

2. Related Works

This section starts by introducing the BERT network model, the baseline work in the area of ASAG, and describing how dataset augmentation influences the performance of ASAG.

In some various NLP tasks, BERT set a new state-of-the-art performance. BERT is a pre-trained transformer network [18]. Reimers and Gurevych recommended Sentence-BERT (SBERT), which uses Siamese BERT-Networks [14] to overcome some deficiencies in BERT. SBERT is a modification of the pre-trained BERT network that uses Siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine similarity.

The SentenceTransformers framework provides various pre-trained models for NLP tasks. The model size and performance are different to each other. All these models have been evaluated for performance sentence embeddings and performance semantic search [14,19]. Table 1 shows some existing models related to sentence classification and question-answering.

Table 1. SentenceTransformers model size.

Model	Model Size
paraphrase-albert-small-v2 (AS)	43 MB
all-MiniLM-L6-v2 (M6)	80 MB
bert-base-uncased (BB)	110 MB
all-MiniLM-L12-v2 (M12)	120 MB
multi-qa-distilbert-cos-v1 (MD)	250 MB
all-distilroberta-v1 (DR)	290 MB
stsb-distilbert-base (SD)	330 MB
multi-qa-mpnet-base-dot-v1 (MM)	420 MB
quora-distilbert-base (QD)	500 MB
Sentence-T5-large (T5)	640 MB
all-roberta-large (RL)	1360 MB
Sentence-T5-XL (TX)	2370 MB

Several studies with varied models have achieved a good performance. Alreheli et al. proposed automatic short answer grading using paragraph vectors and transfer learning embedding [12]. In this work, they utilized the Texas dataset by Mohler to evaluate the models. The input for the ASAG model is the vector that represents the student answer (SA), along with the vector that represents the reference answer (RA). In each experiment, the vectors are inferred using two models; the paragraph vector (PV) model and the transfer learning model. Then, the similarity between SA and RA is measured using the cosine similarity. After that, the computed cosine similarity is used as a feature for a regression model to predict a particular answer score. They evaluate the models by comparing the actual score provided in the dataset, along with the predicted score using two evaluation metrics, the Pearson correlation coefficient and root mean square error (RMSE). For the PV vectors, it achieved 0.401 for the Pearson correlation and 0.893 for the RMSE. For the transfer learning, they applied the Roberta-large and Scibert models. The best accuracy achieved by fine-tuning the Roberta-large model on the domain-specific corpus was 0.620 for the Pearson correlation and 0.777 for the RMSE. This superiority is reasonable, because transformers can learn the context of the words from both directions. On the contrary, the pre-trained paragraph vectors perform better than the trained paragraph vectors on a domain-specific corpus. This indicates that paragraph vectors increase the model's generalizability.

The second approach for improving the performance of ASAG is to use transfer learning and augmentation described by the authors of [13]. They fine-tuned three-sentence transformer models on the SPRAG (Short Programming Related Answer Grading Dataset) corpus with five different augmentation techniques: viz., random deletion, synonym replacement, random swap, back translation, and NLPAug. The SPRAG corpus contains student responses involving keywords and special symbols. The dataset size is 4039 records, and it is a binary classification problem. They experimented with four different data sizes (25%, 50%, 75%, and 100%) with the augmented data to determine the impact of training data on the fine-tuned sentence transformer model. An SBERT architecture with a pretrained language model (PLM) was used for training. The experimentation used the stsb-distilbert-base, paraphrase-albertsmall-v2, and quora-distilbert-base pre-trained sentence transformer models. This paper provides an exhaustive analysis of fine-tuning pretrained sentence transformer models with varying sizes of data by applying text augmentation techniques. They found that applying random swap and synonym replacement techniques together while fine-tuning gave a significant improvement, with a 4.91% increase in accuracy (84.21%) and a 3.36% increase in the F1-score (88.11%).

The third approach, which achieved the best result so far, is integrating transformer-based embeddings and a BI-LSTM network [20]. The proposed model uses pretrained "transformer" models, specifically T5, in conjunction with a BI-LSTM architecture which is effective in processing sequential data by considering the past and future context. This research evaluated several pre-processing techniques and different hyperparameters to

identify the most efficient architecture. Experiments were conducted using a standard benchmark dataset named the North Texas Dataset. This research achieved a state-of-the-art correlation value of 92.5%.

A recent study published in 2024 proposed paraphrase generation and supervised learning for improving ASAG performance [21]. First, they provided a sequence-to-sequence deep learning model that targets generating plausible paraphrased reference answers conditioned on the provided reference answer. Secondly, they proposed a supervised grading model based on sentence-embedding features. The grading model enriches features to improve accuracy, considering multiple reference answers. Experiments are conducted both in Arabic and English. They show that the paraphrase generator produces accurate paraphrases. Using multiple reference answers, the proposed grading model achieves a root mean square error of 0.6955, a Pearson correlation of 88.92% for the Arabic dataset, an RMSE of 0.779, and a Pearson correlation of 73.5% for the English dataset. While fine-tuning pre-trained transformers on the English dataset provided a state-of-the-art performance (RMSE: 0.762), our approach yields comparable results.

Data augmentation has become important in ASAG because more alternative answers can help accommodate the diversity of student answers. Howeve, generating these manually is difficult and needs significant effort. Some suggested methods used for augmenting data in NLPs include random deletion, synonym replacement, random swap, and back translation [13,17]. And in recent years, paraphrase generation become one of the effective strategies in data augmentation. Okur et al. in Ref. [22] used BART and GPT-2 as the paraphrasing model. With the development of GPT, we also think that GPT can be one of the good strategies to generate paraphrasing, especially with the existence of GPT-3.5 or GPT-4.

3. Materials and Methodology

As shown in Figure 2, our proposed method includes processing a dataset, training and fine-tuning a model, and evaluating the model. In this section, we will show the dataset, the pre-processing step for the data, the model implementation for this experiments the use of evaluation metrics, and the overall experimental setup.

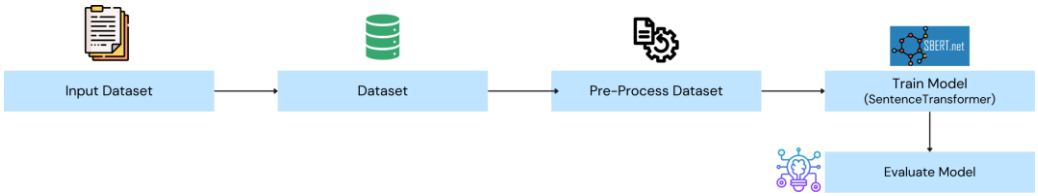


Figure 2. General architecture of proposed method.

For this experiment, we fine-tune the model by hyperparameter optimization. The details of this implementation will be discussed in the next section.

3.1. Dataset

The Mohler dataset comprises questions and answers in an introductory course in computer science provided by Texas University [23]. The goal of the dataset is to evaluate the model in grading the students' answers by comparing them with the evaluator's desired answer. It constitutes 2273 answers from 10 assignments and 2 examinations, collected from 31 students for 80 different questions.

Each answer in the assignment is graded from 0 (not correct) to 5 (totally correct) by two evaluators, who are specialized in the computer science major. The average of the two evaluators' scores is considered as the standard score of each answer. Each answer is graded from 0 to 5, in which grade 0 refers to (wrong), grade 5 refers to (correct), and grades 1 to 4 to partially correct answers. We used the average grade following the original research in this work. We show an example of the dataset in Table 2.

Table 2. Examples of questions and answers for the dataset.

Questions	Desired Answer	Student Answer	Score Avg
What is the role of a prototype program in problem solving?	To simulate the behavior of portions of the desired software product.	A prototype program simulates the behaviors of portions of the desired software product to allow for error checking.	4
		To simulate portions of the desired final product with a quick and easy program that does a small specific job. It is a way to help see what the problem is and how you may solve it in the final project.	5

Figure 3 shows the distribution of each grade label in the Mohler dataset. It can be seen that the grade label classification is not balanced, especially for grade label 0 and grade label 1. Since the amount of data affects the results, these kinds of data also make the performance less good.

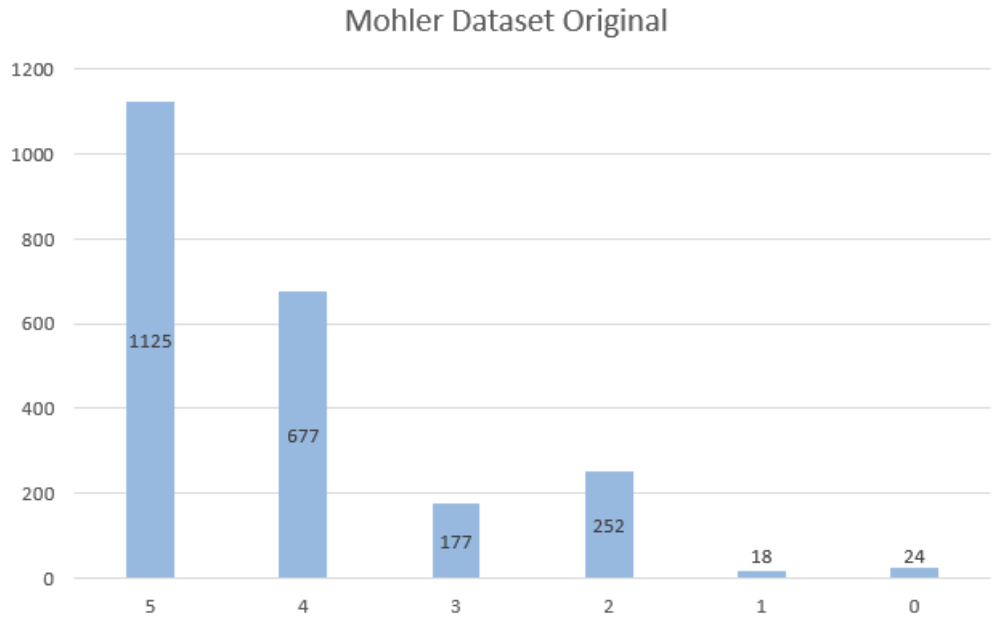


Figure 3. Mohler dataset grade label distribution information.

Bonthu et al. [13] and Ouahrani et al.'s [21] research results show that data augmentation improves the ASAG performance, even if only slightly different. This is the basis for augmenting data so that the dataset become more balanced.

3.2. Data Pre-Processing

Before starting the analysis of the responses, we initially applied some pre-processing steps to remove irrelevant characters (e.g., numbers, punctuation) and turn the text into lowercase. After that, we have conducted only a tokenization. The same as Gaddipati et. al. in [10], we did not use any other checker. Since these transfer learning models are trained on a huge vocabulary, it is plausible to assume that they can understand the misspelled words to an extent. The versatility of transfer learning models to assign an embedding to the new words also assisted in disregarding the spelling mistakes. Other experiments also

applied the removal of stopwords and lemmatization, to check whether the result is better or not.

3.3. Automatic Grading

Based on previous research, several strategies have been implemented to obtain a good ASAG performance. However, the best results from the research pf Gooma et al. [20] used the T5 model to achieve a correlation value of 92.5%. The T5 model has a large model size, as mentioned in Table 1, and is also computationally expensive. This research tries to find the right combination of models and hyperparameters to get better results with lower computational costs. Figure 4 depicts the recommended ASAG process.

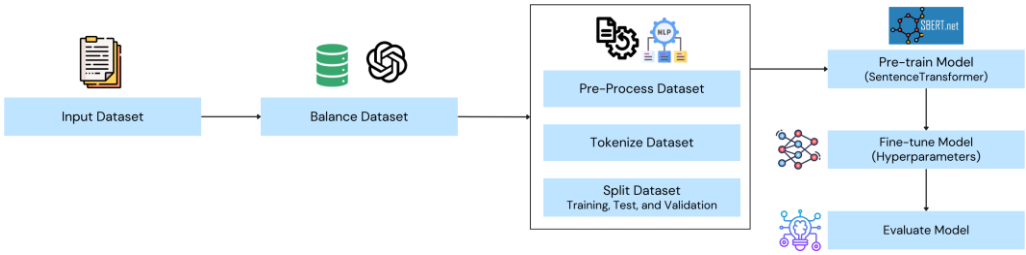


Figure 4. Modification of our proposed method.

This research will utilize eight SentenceTransformers models which have a relatively small size and fine-tune the models using some hyperparameters. Based on Table 1, we recommend several SentenceTransformers models, including paraphrase-albert-small-v2, all-MiniLM-L6-v2, bert-base-uncased, all-MiniLM-L12-v2, multi-qa-distilbert-cos-v1, all-distilroberta-v1, stsb-distilbert-base, and multi-qa-mpnet-base-dot-v1. All of these models are less than 500 MB in size.

In this study, we will fine-tune each model by exploring different combinations of hyperparameters. The parameters used include the size of the training–test data split, the number of epochs, learning rate, pre-processing steps, batch size, and the utilization of gradient checkpointing. Gradient checkpointing serves as a technique aimed at mitigating the memory requirements during deep neural network training, at the cost of having a small increase in computation time [24]. The system will be evaluated using various evaluation metrics, such as accuracy, F1-score, and Pearson correlation. All details about the experimental setup will be explained further in the next sub-section.

3.4. Dataset Balancing

To address the problem of data imbalance, we propose a data augmentation strategy utilizing GPT. The GPT methods used in this research are GPT-3.5 (model gpt-3.5-turbo-1106) and GPT-4 (model gpt-4). The method used is prompt engineering, using GPT to generate new sentences for each class. We implement the prompt engineering in GPT model based on the concept of each grade label-specific characteristics:

- Label 0: generate new opposite sentences of desired answer in dataset.
- Labels 1–4: generate new sentences by paraphrasing the existing student answer. The amount of data depends on the existing amount of data and the maximum amount of data in other labels.
- Label 5: generate new sentences by paraphrasing the existing desired answer.

By constructing appropriate prompts tailored to the paraphrasing task, we leverage the advanced natural language processing capabilities of GPT-3.5 and GPT-4 to generate linguistically diverse and context-specific rephrasings of student answers.

After generating new sentences, we check the quality of the new sentences using METEOR (Metric for Evaluation of Translation with Explicit Ordering) and cosine similarity. Figure 5 shows the augmentation process using GPT; the main idea is obtaining new sentences by paraphrasing the existing sentences.

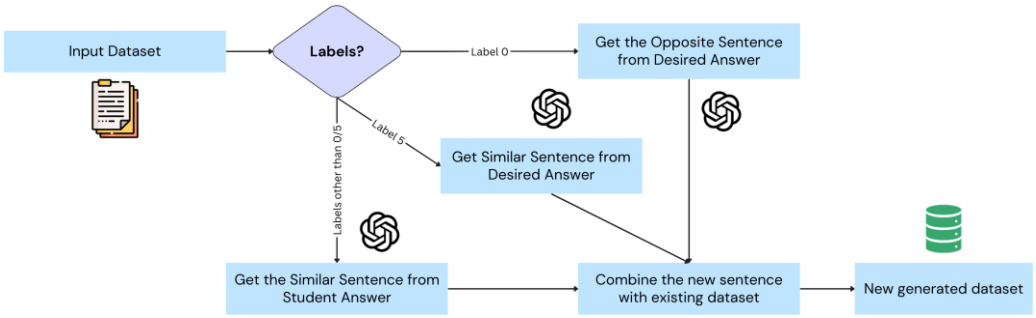


Figure 5. Proposed data augmentation process.

3.5. Evaluation Metrics

All processes need to be evaluated, both the new generated datasets and the short answer grading systems. The evaluation metrics that are commonly used to measure system performance can be seen in the following points.

1. Accuracy

Proportion of correctly graded answers. Accuracy is a straightforward metric; it is essential to consider its limitations, especially in scenarios with imbalanced datasets. In imbalanced datasets, where one class significantly outweighs the other, accuracy can be misleading. To address the limitations of accuracy, we use other evaluation metrics such as precision, recall, and F1-score [25].

2. F1-Score

The F1-score is a weighted comparison of average precision and recall. The formula for the F1-score is as follows:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{1}$$

The F1-score pays attention to the model's ability to handle imbalanced data classes. In addition, by using the F1-score as one of the evaluation metrics, we can compare the resulting model with other published models.

3. Pearson Correlation Score

Pearson's correlation is the most commonly used method in statistics to evaluate the strength and presence of a linear relationship between predicted and manual grades. Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations [20].

4. Root Mean Square Error (RMSE)

Root mean square error or root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions. It shows how far predictions fall from the measured true values using Euclidean distance. The use of RMSE is very common, and it is considered an excellent general-purpose error metric for numerical predictions. RMSE is calculated as follows:

$$RMSE = \sqrt{MSE}, \tag{2}$$

Meanwhile, mean square error (MSE) measures the square of differences between predictions and target values and computes the mean of them. MSE is calculated as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{3}$$

5. METEOR

METEOR stands for Metric for Evaluation of Translation with Explicit Ordering; it is known for its higher correlation with human judgment, especially at the sentence level. These metrics always take a value between 0 and 1. This value indicates how similar the predicted text is to the reference texts, with values closer to 1 representing more-similar texts. METEOR's ability to measure the quality of the generated answer is based on unigram precision and recall. It significantly improves the correlation with human judgments. METEOR computes the similarity score of two texts by using a combination of unigram precision, unigram recall, and some additional measures like stemming and synonymy matching [21].

The cosine similarity technique is used for measuring the similarity between two vectors. The way it works is by measuring the cosine of the angle between two documents which are expressed in vectors. Haskova et al. stated that the angle between vectors determines whether they are pointing in the same or different directions. If vectors are pointing in the same direction, it means that the documents are similar; the closer they are expressed on the axis, the more similar they are. Vice versa, the farther they are expressed on the axis, the less similar they are [26].

3.6. Experimental Setup

As we ran many SBERT models and it required a more powerful graphical processor unit (GPU), we used Google Colab's T4 GPU with a high RAM (around 52 GB). Our study was based on the Hugging Face model based on Transformers for PyTorch 1.11.0 and TensorFlow 2.0. We also employed the OpenAI API, leveraging prompt-engineering techniques with GPT-3.5 and GPT-4, to paraphrase sentences for the data augmentation process.

Initially, the general scenario that was formed was to compare the implementation of each model by fine-tuning the various hyperparameters mentioned previously. We tried with epoch values of 8, 10, 12, or 16, then batch sizes of 8, 16, or 32, and learning rate values of 5×10^{-5} or 5×10^{-6}. We experimented with various scenarios that were combinations of these hyperparameters with the original dataset and the models mentioned. Based on these initial experiments, the best result was obtained from using epoch = 12, batch size = 16, and learning rate = 5×10^{-5}. So, the next experiment would use these three fixed parameters and the other parameters combined. These included whether or not to apply removing stopwords, whether to apply gradient checkpoints or not, and dataset-splitting sizes.

The next scenario was related to the data augmentation process. Initially, we used the GPT-3.5 model with 2 different temperature values. "Temperature" refers to a parameter that influences the randomness degree of the generated text. A low temperature (close to 0) leads to more deterministic outputs, where the model tends to choose words with higher probabilities, resulting in a more conservative and repetitive text. A high temperature (with a maximum value = 2) increases randomness in the generated text, causing the model to sample from less predictable words, resulting in a more diverse but potentially less coherent text. The results for these new generated sentences were evaluated using METEOR and cosine similarity, giving better results for temperature = 0.7. Next, with the parameter value temperature = 0.7, we also tried generating a new sentence with GPT-4. So, we will compare the results of generated new sentences between GPT-3.5 and GPT-4.

In the next scenario, we will compare the performance of the grading system using the dataset before augmentation and after the augmentation process. In addition, we also have a scenario for conducting experiments using larger SentenceTransformers models. Then the results will be compared between our recommended model and the larger model. The evaluation metrics used are as mentioned in the previous section. We will also check the running time of each model.

4. Result and Discussion

In this session, we will explain and discuss the experiment results based on the experimental setup described previously.

4.1. Initial Answer-Grading Process

As mentioned before, we will conduct experiments using the original dataset and some fixed parameters include epoch = 12, batch size = 16, and learning rate = 5×10^{-5}. The results shown in Table 3 include the best combination of hyperparameters from the overall results. We split the dataset into an 80–20 ratio or 70–30 ratio for training and testing data, respectively. The pre-processing step used for this experiment only removed special characters and changed the sentences into lowercase.

Table 3. Model performance for original dataset.

Model Name	Gradient Checkpointing	Split Size	RMSE	F1-Score	Accuracy	Pearson Correlation
bert-base-uncased	Yes	70–30	1.61902	0.19927	0.22287	0.21246
stsb-distilbert-base			1.04117	0.65803	0.66862	0.63649
all-distilroberta-v1			1.06047	0.66809	0.64516	0.66427
all-MiniLM-L6-v2			0.99158	0.60616	0.64367	0.64367
multi-qa-mpnet-base-dot-v1			0.89391	0.70164	0.70614	0.45196
multi-qa-distilbert-cos-v1			1.08879	0.63096	0.64474	0.47758
paraphrase-albert-small-v2			1.31466	0.58836	0.58333	0.43599
all-MiniLM-L12-v2			0.99299	0.63995	0.65351	0.43184
bert-base-uncased	No	80–20	1.91571	0.42417	0.40351	0.25566
stsb-distilbert-base			1.31803	0.67083	0.67105	0.59558
all-distilroberta-v1			1.24106	0.73221	0.72807	0.65142
all-MiniLM-L6-v2			1.19878	0.58192	0.56579	0.64576
multi-qa-mpnet-base-dot-v1			1.22907	0.68833	0.68859	0.76269
multi-qa-distilbert-cos-v1			1.4047	0.67267	0.67105	0.71517
paraphrase-albert-small-v2			1.79938	0.50837	0.49123	0.62883
all-MiniLM-L12-v2			1.15167	0.63541	0.64474	0.71271

Based on the results in Table 3, the all-distilroberta-v1 and mul-ti-qa-mpnet-base-dot-v1 models show promising results, although their performance across all evaluation metrics is still not better than the existing research. The best RMSE value obtained is nearly 0.9, whereas even the smallest value in the previous research reached 0.77. Moreover, the F1-score, accuracy and Pearson correlation values are only around 0.7, while the previous research has achieved more than that. So, we conducted additional experiments using a new dataset that has undergone a data augmentation process with GPT.

4.2. Dataset Balancing by Data Augmentation with GPT

Based on the previously explained scenario, the process of balancing the dataset with data augmentation will use a parameter value of temperature = 0.7 and will also utilize both the GPT-3.5 and GPT-4 models. For grade label = 0, the prompt text used will be "Please make a completely different sentence from this following sentence: '{answer}' so it counts as an opposite sentence" to get the opposite or antonyms. Meanwhile, for other grade labels, the prompt text used will be "Please paraphrase the following sentence '{answer}'" to get similar sentences or synonyms. Figure 6 shows the new dataset, with additional data generated by the GPT-3.5 model. Subsequently, the results of this process will be referred to as BalMohler-3.5, which means Balanced Mohler dataset with GPT-3.5.



Figure 6. New generated dataset using GPT-3.5.

Figure 7 depicts the new dataset with added data resulting from data augmentation using GPT-4. Compared to Figure 3, this new dataset has more evenly distributed numbers. The standard deviation for the original Mohler dataset is 400, while the standard deviation for the new dataset is around 98 for both models. This dataset will be referred to as BalMohler-4.

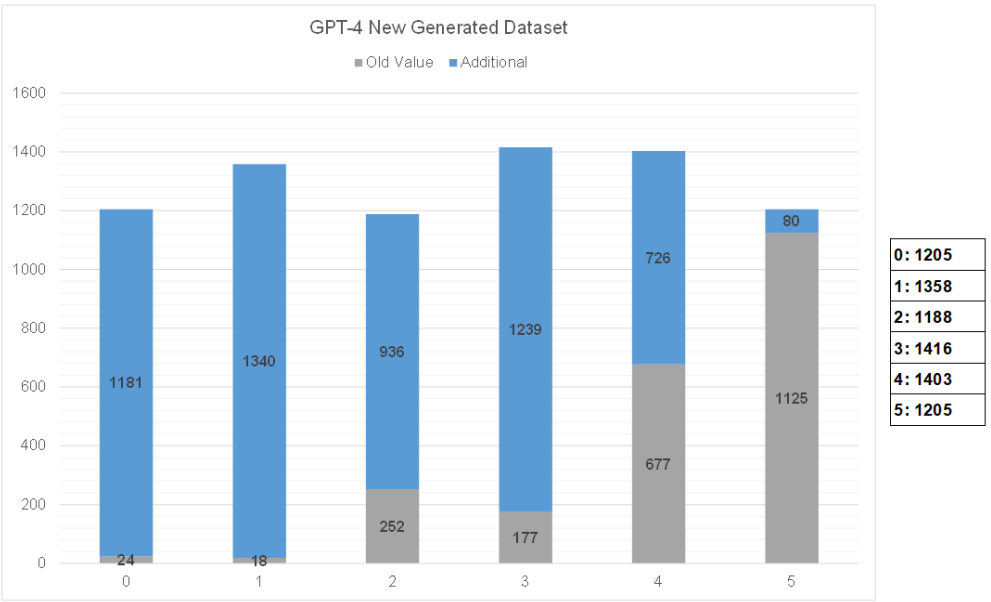


Figure 7. New generated dataset using GPT-4.

In addition to the distribution of data, we will also evaluate the results of these new generated sentences based on METEOR and cosine similarity scores. To facilitate evaluation, we will categorize grade labels into two categories: the correct category for grade labels 2–5 and the false category for grade labels 0–1. The grade labels 0–1 are considered as the false category because they represent answers that are far from the correct answer. Table 4 shows the evaluation results of data augmentation from both models.

Table 4. Data augmentation result evaluation.

Augmentation Model	Grade Category	Cosine Similarity	METEOR
GPT-3.5	False	0.4369	0.59722
GPT-3.5	Correct	0.8040	
GPT-4	False	0.3645	0.59309
GPT-4	Correct	0.8080	

These new generated sentences will be considered as student answers and will be compared with the desired answers for evaluation. So, a smaller value in the cosine similarity score indicates better results for the false category. Meanwhile, for the correct category, a larger value indicates better performance. The new sentences of the GPT-4 model are better, although there is only a small increase. Meanwhile, the METEOR score reflects the overall text quality, where a higher score indicates better quality. As seen in Table 4, the results of the new generated sentences with the GPT-3.5 model show a better performance, although the difference is not significant compared to the results from the GPT-4 model. Therefore, for future research, we will continue to evaluate the ASAG process using both of these new generated datasets.

4.3. Answer-Grading Process after Balancing Dataset

These experiments will use a new dataset with the additional data through data augmentation with the GPT-3.5 and GPT-4 models. We will conduct various experiments with existing model combinations, while fine-tuning the hyperparameters. From the several experiments that have been carried out, we will present the best results among them. The details of the scenarios conducted can be seen in Table 5. Exp 1 and Exp 5 represent the best scenarios for experiments using the original Mohler dataset, for which the results can be seen in Table 3. Removing stopwords is not included in the parameter combination because the experimental results are not significant, so it was not involved.

Table 5. Detail experiment scenario.

Experiment	Pre-Processing	Gradient Checkpointing	Split Size
Exp 1	SpCh LC	Yes	70–30
Exp 2	SpCh LC	No	70–30
Exp 3	SpCh LC	Yes	80–20
Exp 4	SpCh LC RS	Yes	80–20
Exp 5	SpCh LC	No	80–20

Each scenario is executed with a fixed parameters set based on the results of previous experiments. The results of this experiment will be evaluated using the evaluation metrics mentioned earlier, namely RMSE, F1-score, accuracy, and Pearson correlation. These experiments include the SentenceTranformers model mentioned in Table 1.

Figure 8 displays the RMSE values from our eight recommended models when implemented on three datasets: Original Mohler, BalMohler-3.5, and BalMohler-4, represented as Ori, Aug-3.5, and Aug-4, respectively, for each subsequent figure. Generally, better results are obtained from using BalMohler-3.5 and implementing the all-distilroberta-v1 model. To shorten the model names, what is shown in the figure is the abbreviation for each model based on Table 1.

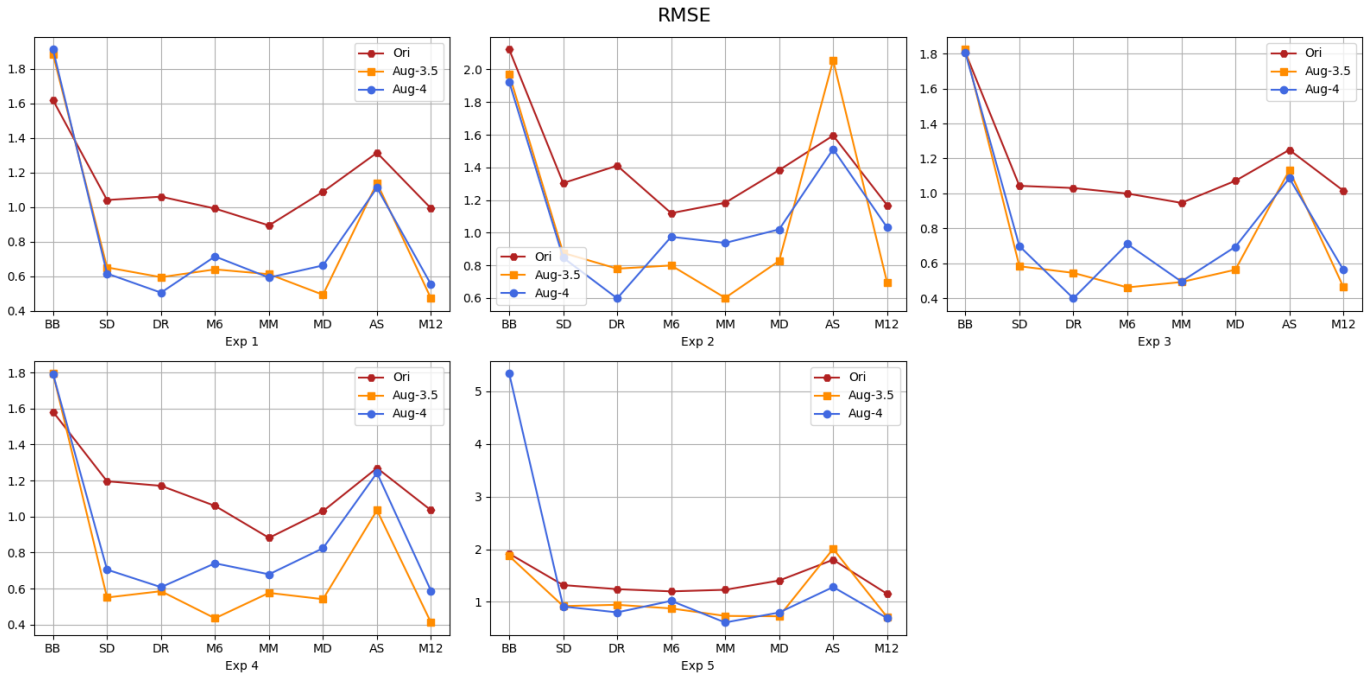


Figure 8. Comparing RMSE for all models and dataset.

The best RMSE score of 0.39913 was obtained from the implementation of Exp 3, using the all-distilroberta-v1 model with a new balanced dataset from data augmentation with the GPT-4 model. However, in general, using BalMohler-3.5 results in a smaller average RMSE score. Meanwhile, the worst RMSE scores were obtained from implementing Exp 5 on the bert-base-uncased model with BalMohler-4, with a value of 5.33808.

Figure 9 displays the experimental results based on the F1-score score obtained. Overall, balancing the dataset by implementing data augmentation improves the performance of the grading system.

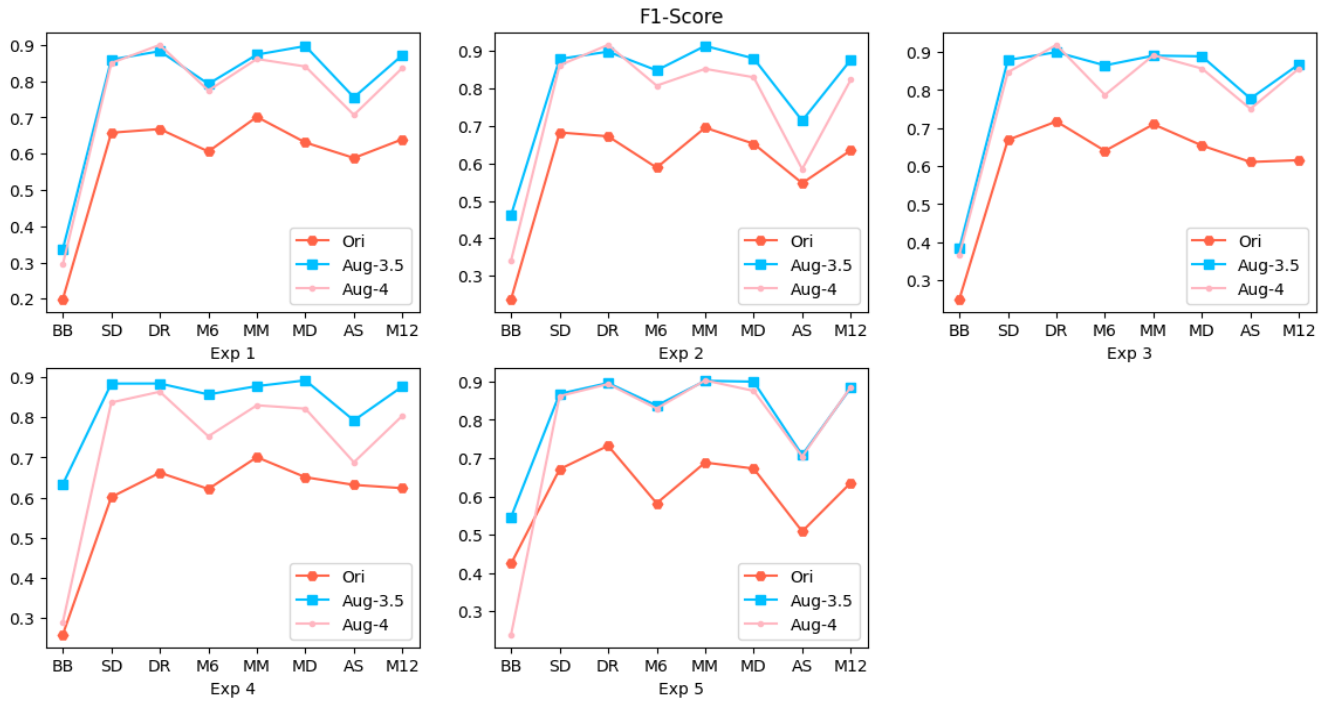


Figure 9. Comparing F1-score for all models and datasets.

The best F1-score result of 0.91886 was obtained from Exp 3 with BalMohler-4 and the implementation of the all-distilroberta-v1 model. Similar to the RMSE results, on average, the F1-score results from using BalMohler-3.5 are better than BalMohler-4. The worst result, 0.2391, comes from Exp 5, using the BalMohler-4 dataset and the bert-base-uncased model. This worst value is close to the worst F1-score value with the Original Mohler dataset (0.2379).

Figure 10 displays the performance evaluation in terms of accuracy. Similar to the other evaluation metrics, the resulting balanced dataset from implementing GPT data augmentation generally improves the performance of the grading system.

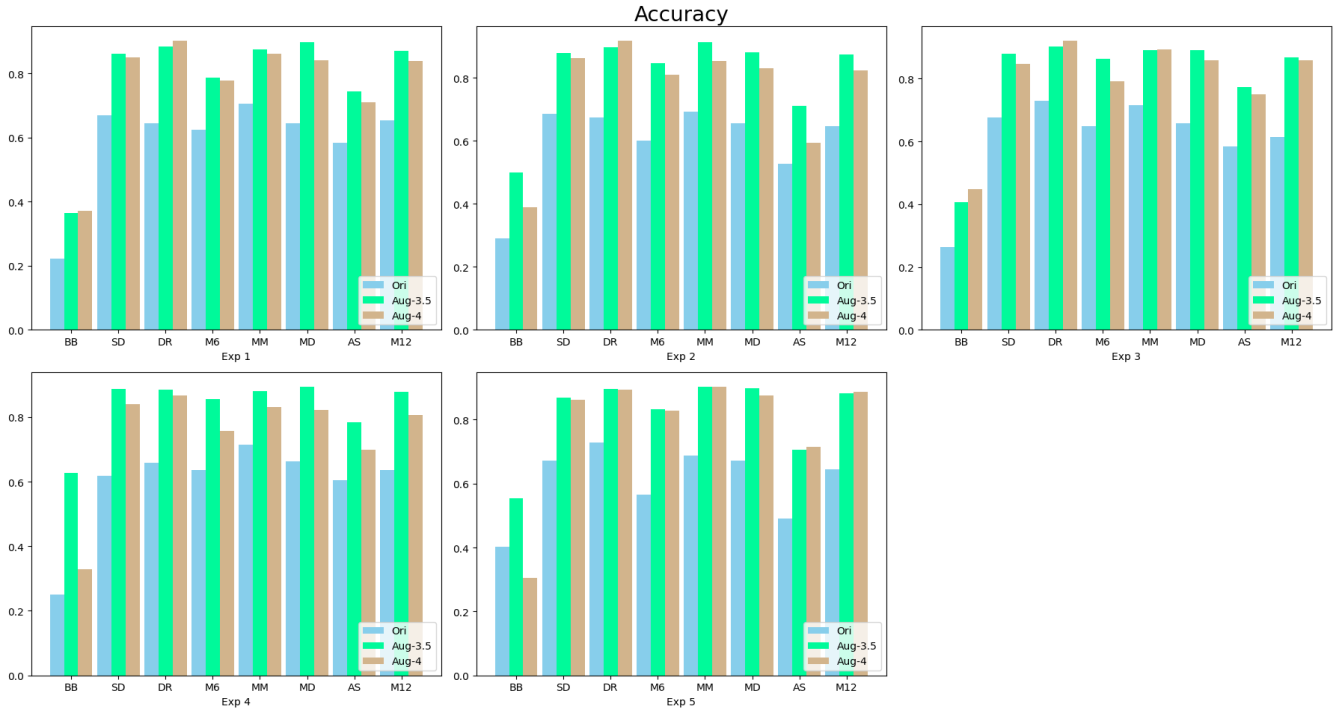


Figure 10. Comparing accuracy for all models and dataset.

The best accuracy result of 0.91969 was obtained from Exp 3 with the implementation of the BalMohler-4 dataset and the all-distilroberta-v1 model. On average, the accuracy value of BalMohler-3.5 implementation is also better than the implementation of BalMohler-4. The lowest accuracy score, 0.30488, comes from Exp 5, with the implementation of the BalMohler-4 dataset and the bert-base-uncased model.

Figure 11 displays the final evaluation results based on the Pearson correlation value. It is also clear that the balanced dataset from the implementation of GPT data augmentation improves performance results.

Slightly different from the previous evaluation, the best result was obtained from Exp 3 and the all-distilroberta-v1 model but with the implementation of BalMohler-3.5. The highest Pearson correlation score is 0.95855. The lowest score, 0.28748, was obtained from Exp 4 with the implementation of the BalMohler-4 dataset and the bert-base-uncased model.

Based on the experimental results so far, on average, the best implemented model is the all-distilroberta-v1 model, along with the use of BalMohler-3.5. The all-distilroberta-v1 model has a size of around 290 MB and can achieve good results for all the evaluation metrics conducted. As mentioned in Section 2, there have been other studies that have succeeded in achieving satisfactory evaluation metrics values, but those studies typically use larger models than those we recommend. For instance, Alreheli and Alghamdi used the all-roberta-large model [12], which is nearly 400% larger than our recommended model, and Gomaa used T5-XL [20] which is almost eight times larger than our recommended model.

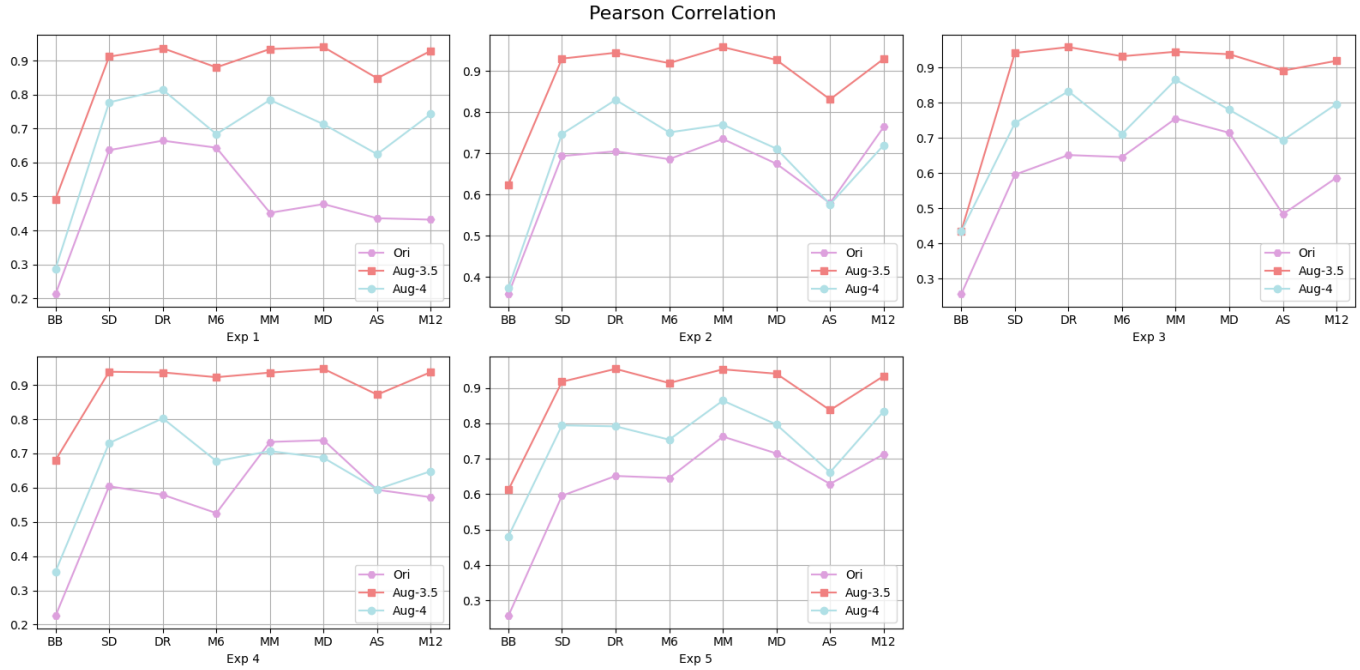


Figure 11. Comparing Pearson correlation for all models and datasets.

We also conducted additional experiments using larger models to observe their performance. Table 6 displays our overall best experimental results, as well as the results from previous research. Based on this summary, it can also be seen that data augmentation implementation to balance the dataset helps improve the performance for smaller-sized SentenceTransformers models. A smaller size means a smaller number of parameters and also results in a faster processing time [14,27].

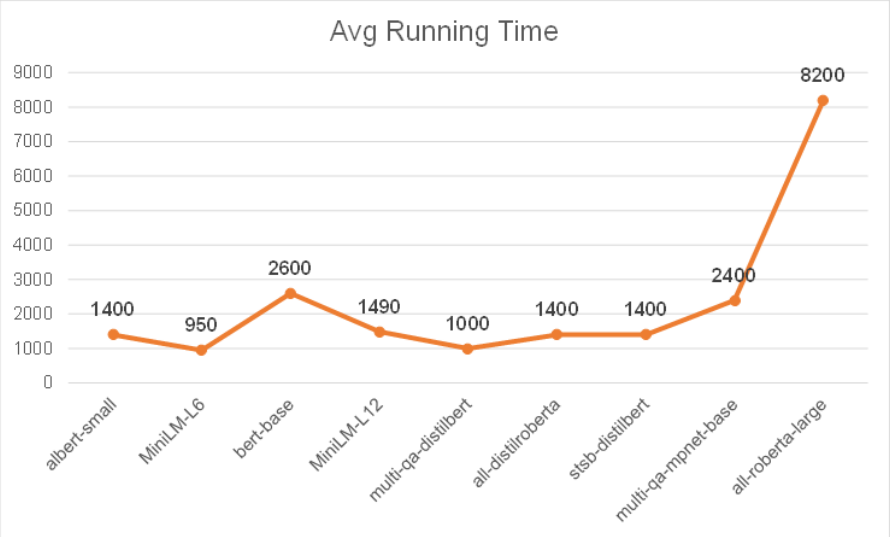
Table 6. Comparison of all experimental results with previous research.

Model Name	Dataset	Pre-Processing	Gradient Checkpointing	Split Size	Batch Size	RMSE	F1-Score	Accuracy	Elation
all-distilroberta-v1	Mohler	SpCh + LC	No	80–20 $^{+}$	16	1.2411	0.7322	0.7281	0.6514
multi-qa-mpnet-base-dot-v1						1.2291	0.6883	0.6886	0.7627
all-distilroberta-v1	BalMohler-3.5	SpCh + LC	Yes	80–20 $^{\#}$	16	0.5449	0.8998	0.9009	0.9586
multi-qa-mpnet-base-dot-v1						0.4925	0.8907	0.8912	0.9454
multi-qa-mpnet-base-dot-v1			No	70–30 *		0.6012	0.9139	0.9145	0.9583
all-distilroberta-v1	BalMohler-4	SpCh + LC	Yes	80–20 $^{\#}$	16	0.3991	0.9189	0.9197	0.8329
multi-qa-mpnet-base-dot-v1						0.4954	0.8917	0.8929	0.8657
RoBERTa-large	BalMohler-4 BalMohler-3.5	SpCh + LC	Yes	80–20 $^{\#}$	16	0.4021	0.9374	0.9377	0.9612
						0.5096	0.9328	0.9334	0.9579
RoBERTa-large [12]	Mohler	SpCh + RS	-	80–20	16	0.777	-	-	0.620
T5-XL [20]	Mohler	LC	-	80–20	3	0.109	-	-	0.928
paraphrase-albert-small-v2 [13]	SPRAG + Aug	SpCh	-	85–15	16	-	0.8811	0.8421	-
Ridge-LR [21]	Mohler + Aug-E	SpCh + RS	-	70–30	-	0.779	-	-	0.735
Ridge-LR [21]	Mohler + Aug-A	SpCh + RS	-	70–30	-	0.6955	-	-	0.8892

* Exp 2, $^{\#}$ Exp 3, $^{+}$ Exp 5.

Our best results from the experiment are marked with green in the respective column. Results labeled in blue indicate additional experiments using larger models. The results obtained for F1-score, accuracy, and Pearson correlation are indeed better than those from experiments with smaller models, but the difference in performance is not significant compared to the average running time. In Figure 12 below, it can be seen that the average running time of the all-roberta-large model is many times longer than the other models. Given the disparity in performance, it is not proportional to the computational cost required. Balancing the dataset is also recommended in research by Bonthu et al. [13] and Ouahrani

et al. [21]. There is an increase in performance when using the augmented data, which is also consistent with our experimental results.



Model	Model Size (MB)
paraphrase-albert-small-v2	43
all-MiniLM-L6-v2	80
bert-base-uncased	110
all-MiniLM-L12-v2	120
multi-qa-distilbert-cos-v1	250
all-distilroberta-v1	290
stsb-distilbert-base	330
multi-qa-mpnet-base-dot-v1	420
all-roberta-large	1360

Figure 12. Average running time of each experiment.

The RMSE scores obtained from our experiments are indeed larger than those from Gomaa's research [20], but that does not have much impact because of other better evaluation metrics values. Moreover, the use of the T5-XL model would definitely require even larger and more expensive resources. The comparative F1-score and accuracy values were only obtained from research by Bonthu et al. [13], and our best experimental results on average also exceed those results. Bonthu et al. used the relatively small paraphrase-albert-small-v2 model, which is only about 40 MB in size. However, our recommended model can potentially perform better, due to differences in the datasets. Bonthu et al. used the SPRAG dataset, which is a binary classification problem [13], while the Mohler dataset consists of grade labels ranging from 0 to 5. With a more complex dataset, even though it requires a larger model, the results can still compete with simpler models. Note also that the paraphrase-albert-small-v2 model has an average running time that is not significantly different from the all-distilroberta-v1 model, which is about seven times larger in size. The Pearson correlation scores from our recommended model can also achieve better results than the existing research. When the same dataset and the same hyperparameter fine-tuning process are implemented on larger models, it indeed produces better results, but this performance improvement is not proportional to the larger and more expensive computational cost.

This research aims to find a simpler model with a proper fine-tuning process. The experiments have shown that relatively smaller-sized models with the proper fine-tuning can achieve a good performance. The data augmentation for balancing the dataset itself also contributes to a significant improvement in the performance results of this grading system. The all-distilroberta-v1 model, which is less than 300 MB in size, with the proper hyperparameter selection and combined with balancing the dataset, can compete with the results of larger and more complex models.

4.4. Additional Experiments

We conducted additional experiments to determine whether our proposed method also improves the performance of the ASAG system on other datasets. We utilized the SemEval-2013 dataset, a benchmark dataset from the SemEval-2013 Shared Task 7 [28]. Specifically, we used the two-way SciEnts Bank subset, which includes two grade labels: "correct" as grade label 1 and "incorrect" as grade label 0. This dataset consists of questions, desired answers, student answers, and two-way grade labels in the science domain.

We used both the original dataset and an augmented version. The initial dataset contained 4925 rows, with 2944 rows (60%) labeled as grade 0 and 1981 rows (40%) labeled as grade 1. We applied the same augmentation process to this dataset. Using prompt engineering in GPT models, we generated additional answers for grade label 0 by using antonyms of the desired answer and synonyms of words from the student answers to create new answers for grade label 1. Figure 13 illustrates the data distribution for both the original and the balanced datasets after augmentation with the GPT-3.5 and GPT-4.0 models. The blue bar represents data with a grade label of 0, while the orange bar represents data with a grade label of 1.

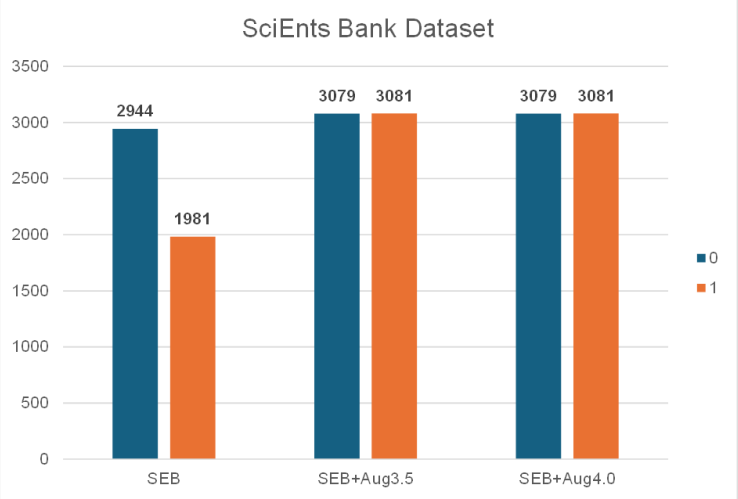


Figure 13. SciEnts Bank datasets data distribution.

We also evaluated the newly generated dataset using the METEOR score and cosine similarity score. The evaluation results are presented in Table 7, which shows that the dataset generated using the GPT-3.5 model performed better. These datasets were also implemented using the recommended models from the experiments in the previous section: the all-distilroberta-v1 and multi-qa-mpnet-base-dot-v1 models.

Table 7. SciEnts Bank data augmentation results evaluation.

Augmentation Model	Grade Category	Cosine Similarity	METEOR
GPT-3.5	False	0.4411	0.5921
GPT-3.5	Correct	0.7121	
GPT-4	False	0.4429	0.5498
GPT-4	Correct	0.7059	

From the five scenarios mentioned in Table 5, we only present the best two results for this additional experiment, using the same combination of hyperparameters. The results are displayed in Table 8.

Our best results from these additional experiments are also highlighted in green in the respective column. These results indicate that the augmentation process successfully increased system performance. Moreover, the results labeled in blue indicate experiments using larger models. The larger models achieved better scores in F1-score, accuracy, and Pearson correlation but, as with the previous dataset, the performance improvement was not significant compared to the average running time. When compared with previous research using the same dataset, our proposed method also improves performance quite well.

Table 8. Comparison of additional experimental results with previous research.

Model Name	Dataset	Pre-Processing	Gradient Checkpointing	Split Size	Batch Size	RMSE	F1-Score	Accuracy	Pearson Correlation	Average Runtime
all-distilroberta-v1	SEB	SpCh + LC	Yes	80–20 #	16	0.8857	0.8186	0.8236	0.6399	1066 s
	SEB + Aug3.5					0.8205	0.8556	0.8556	0.7103	1406 s
	SEB + Aug4					0.7429	0.8375	0.8377	0.6738	1390 s
multi-qa-mpnet-base-dot-v1	SEB	SpCh + LC	Yes	80–20 #	16	0.7818	0.8301	0.8296	0.6559	1806 s
	SEB + Aug3.5					0.7329	0.8507	0.8507	0.6999	2285 s
	SEB + Aug4					0.7956	0.8392	0.8393	0.6765	2285 s
RoBERTa-large	SEB	SpCh + LC	Yes	80–20 #	16	0.8648	0.8722	0.8722	0.7411	6381 s
	SEB + Aug3.5					1.0093	0.8657	0.8658	0.7319	6769 s
all-distilroberta-v1	SEB	SpCh + LC	No	80–20 +	16	1.1348	0.8358	0.8357	0.6668	909 s
	SEB + Aug3.5					1.1527	0.8313	0.8312	0.6582	1114 s
	SEB + Aug4					1.1986	0.8211	0.8215	0.6404	1121 s
multi-qa-mpnet-base-dot-v1	SEB	SpCh + LC	No	80–20 +	16	0.8226	0.8518	0.8519	0.6928	1898 s
	SEB + Aug3.5					0.8161	0.8639	0.8636	0.7265	2409 s
	SEB + Aug4					0.9179	0.8521	0.8523	0.7058	2381 s
Graph Convolutional Networks [29]	SEB	-	-	90–10	32	-	0.705	0.710	-	-
	SEB + BT	-	-	90–10	32	-	0.725	0.732	-	-
XLNET base [3]	SEB	-	-	90–10	16	-	0.693	0.702	-	-

Exp 3, + Exp 5, BT: back translation.

When compared with the Mohler dataset, the improvement in experiments with the SciEnts Bank dataset was not very significant. Based on further observations, we found that the Mohler dataset had an average of 20 words per row of data, while the SciEnts Bank dataset had an average of only 12 words per row of data. This difference in word count may contribute to the less significant performance improvement, as fewer words were considered in the evaluation.

5. Conclusions

In this study, we proposed a simpler SentenceTransformers model combined with balancing the dataset and fine-tuning the hyperparameters of the model to handle an automatic short answer grading system. Our recommended SentenceTransformers model has a relatively small size, resulting in manageable resource requirements. We also balanced the dataset using GPT data augmentation, employing prompt engineering in the GPT model to generate new sentences based on existing student answers or desired answers. Additionally, we also fine-tuned the model by combining appropriate hyperparameters to achieve optimal grading performance. From the experiments conducted, the new balanced dataset significantly improved the performance of the grading system, as observed through RMSE, F1-score, accuracy, and Pearson correlation metrics.

The newly generated answer data from GPT also display satisfactory results, with a cosine similarity score reaching 0.8 for the correct category and 0.3 for the false category. This data augmentation aims to create a more balanced distribution of grade labels in the dataset. The implementation of this new balanced dataset also resulted in a significant performance improvement. The best result we obtained reached a Pearson correlation value of 0.9586 from the implementation of the all-distilroberta-v1 model. This model has a relatively small size and underwent a fine-tuning of hyperparameters, as well as the utilization of the new balanced dataset. Key hyperparameters include (1) the use of gradient checkpointing to reduce memory consumption; (2) the split-size ratio for the training and testing datasets, with 80% for training and 20% for testing; (3) a pre-processing step involving the removal of special characters and converting text to lowercase. Other parameters remained fixed across all the experiments, as mentioned earlier. Furthermore, the RMSE, F1-score, and accuracy score also consistently achieved better results compared to the previous research. This has also been demonstrated by additional experimental results. Although the performance increase for the new dataset is not very significant, there is still an improvement. This difference can be attributed to the varying characteristics of the datasets themselves.

In terms of future works, there are several areas that might be explored further. Currently, the dataset used consists only of English-language data. Hence, future research

could explore the implementation of models trained on datasets from other languages. GPT has proven to be able to generate new sentences to enhance performance. Therefore, it is also possible to leverage GPT for language translation when using datasets other than in English.

Author Contributions: Conceptualization, M.C.W. and H.-S.Y.; methodology, M.C.W.; software, M.C.W.; validation, H.-S.Y.; formal analysis, M.C.W.; investigation, M.C.W.; resources, M.C.W.; data curation, M.C.W.; writing—original draft preparation, M.C.W.; writing—review and editing, M.C.W. and H.-S.Y.; visualization, M.C.W.; supervision, H.-S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially supported by a Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure, and Transport (Grant RS-2022-00143782).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wijanto, M.C.; Karnalim, O.; Tan, R. Work in progress: High School Students' Perspective on Assessment Question Types during Online Learning—Preliminary Study for Automated Assessments of Open-ended Questions. In Proceedings of the EDUNINE 2022—6th IEEE World Engineering Education Conference: Rethinking Engineering Education after COVID-19: A Path to the New Normal, Santos, Brazil, 13–16 March 2022; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2022. [CrossRef]
2. Süzen, N.; Gorban, A.N.; Levesley, J.; Mirkes, E.M. Automatic short answer grading and feedback using text mining methods. *Procedia Comput. Sci.* **2020**, *169*, 726–743. [CrossRef]
3. Ghavidel, H.; Zouaq, A.; Desmarais, M. Using BERT and XLNET for the Automatic Short Answer Grading Task. In Proceedings of the CSEDU 2020—Proceedings of the 12th International Conference on Computer Supported Education, Prague, Czech Republic, 2–4 May 2020; SciTePress: Setúbal, Portugal, 2020; pp. 58–67. [CrossRef]
4. Bagaria, V.; Badve, M.; Beldar, M.; Ghane, S. An Intelligent System for Evaluation of Descriptive Answers. In Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020, Thoothukudi, India, 3–5 December 2020; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2020; pp. 19–24. [CrossRef]
5. Tulu, C.N.; Ozkaya, O.; Orhan, U. Automatic Short Answer Grading with SemSpace Sense Vectors and MaLSTM. *IEEE Access* **2021**, *9*, 19270–19280. [CrossRef]
6. Burrows, S.; Gurevych, I.; Stein, B. The Eras and Trends of Automatic Short Answer Grading. *Int. J. Artif. Intell. Educ.* **2014**, *25*, 60–117. [CrossRef]
7. Zhang, L.; Huang, Y.; Yang, X.; Yu, S.; Zhuang, F. An automatic short-answer grading model for semi-open-ended questions. *Interact. Learn. Environ.* **2019**, *30*, 177–190. [CrossRef]
8. Wijaya, M.C. Automatic Short Answer Grading System in Indonesian Language Using BERT Machine Learning. *Rev. d'Intelligence Artif.* **2021**, *35*, 503–509. [CrossRef]
9. Condor, A.; Litster, M.; Pardos, Z. Automatic short answer grading with SBERT on out-of-sample questions. In Proceedings of the 14th International Conference on Educational Data Mining (EDM 2021), Online, 29 June–2 July 2021; International Educational Data Mining Society: Paris, France, 2021; pp. 345–352.
10. Gaddipati, S.K.; Nair, D.; Plöger, P.G. Comparative Evaluation of Pretrained Transfer Learning Models on Automatic Short Answer Grading. *arXiv* **2020**, arXiv:2009.01303.
11. Poulton, A.; Eliens, S. Explaining transformer-based models for automatic short answer grading. In Proceedings of the ACM International Conference Proceeding Series, Association for Computing Machinery, Busan, Republic of Korea, 15–17 September 2021; pp. 110–116. [CrossRef]
12. Alreheli, A.S.; Alghamdi, H.S. Automatic Short Answer Grading Using Paragraph Vectors and Transfer Learning Embeddings. *J. King Abdulaziz Univ. Comput. Inf. Technol. Sci.* **2022**, *11*, 25–31. [CrossRef]
13. Bonthu, S.; Sree, S.R.; Prasad, M.K. Improving the performance of automatic short answer grading using transfer learning and augmentation. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106292. [CrossRef]
14. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv* **2019**, arXiv:1908.10084.

15. Ndukwe, I.G.; Amadi, C.E.; Nkomo, L.M.; Daniel, B.K. Automatic Grading System Using Sentence-BERT Network. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2020; pp. 224–227. [CrossRef]
16. Dzikovska, M.O.; Nielsen, R.D.; Brew, C. Towards Effective Tutorial Feedback for Explanation Questions: A Dataset and Baselines. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Montreal, BC, Canada, 3–8 June 2012; pp. 200–210.
17. Wiratmo, A.; Nopember, I.T.S.; Fatichah, C. Indonesian Short Essay Scoring Using Transfer Learning Dependency Tree LSTM. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 278–285. [CrossRef]
18. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
19. Reimers, N. Sentence Transformers Documentation. Available online: https://www.sbert.net/docs/pretrained_models.html (accessed on 18 January 2024).
20. Gomaa, W.H.; Nagib, A.E.; Saeed, M.M.; Algarni, A.; Nabil, E. Empowering Short Answer Grading: Integrating Transformer-Based Embeddings and BI-LSTM Network. *Big Data Cogn. Comput.* **2023**, *7*, 122. [CrossRef]
21. Ouahrani, L.; Bennouar, D. Paraphrase Generation and Supervised Learning for Improved Automatic Short Answer Grading. *Int. J. Artif. Intell. Educ.* **2024**, 1–44. [CrossRef]
22. Okur, E.; Sahay, S.; Nachman, L. Data Augmentation with Paraphrase Generation and Entity Extraction for Multimodal Dialogue System. In Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), Marseille, France, 21–23 June 2022; European Language Resources Association (ELRA): Paris, France, 2022; pp. 4114–4125.
23. Mohler, M.; Bunescu, R.; Mihalcea, R. Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, OR, USA, 19–24 June 2011; Association for Computational Linguistics: Stroudsburg, PA, USA, 2011; pp. 752–762.
24. Chen, T.; Xu, B.; Zhang, C.; Guestrin, C. Training Deep Nets with Sublinear Memory Cost. *arXiv* **2016**, arXiv:1604.06174.
25. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016.
26. Januzaj, Y.; Luma, A. Cosine Similarity—A Computing Approach to Match Similarity between Higher Education Programs and Job Market Demands Based on Maximum Number of Common Words. *Int. J. Emerg. Technol. Learn.* **2022**, *17*, 258–268. [CrossRef]
27. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. In Proceedings of the 5th EMC2—Energy Efficient Machine Learning and Cognitive Computing, Vancouver, BC, Canada, 13 December 2019.
28. Dzikovska, M.O.; Nielsen, R.; Brew, C.; Leacock, C.; Giampiccolo, D.; Bentivogli, L.; Clark, P.; Dagan, I.; Dang, H.T. SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge. In Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, GA, USA, 14–15 June 2013; Association for Computational Linguistics: Stroudsburg, PA, USA, 2013; pp. 263–274.
29. Tan, H.; Wang, C.; Duan, Q.; Lu, Y.; Zhang, H.; Li, R. Automatic short answer grading by encoding student responses via a graph convolutional network. *Interact. Learn. Environ.* **2020**, *31*, 1636–1650. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.