Cross-Prompt Based Automatic Short Answer Grading System

Lucia D. Krisnawati[1*], Aditya W. Mahastama[2], and Su Cheng Haw[3]
[1,2]Informatics Department, Faculty of Information Technology, Universitas Kristen Duta Wacana
Yogyakarta 55224, Indonesia
[3]Faculty of Computing and Informatics, Multimedia University
Cyberjaya 63100, Malaysia
Email: [1]krisna@staff.ukdw.ac.id, [2]mahas@staff.ukdw.ac.id, [3]sucheng@mmu.edu.my

***Abstract*—Research on Automatic Short Answer Grading (ASAG) has shown promising results in recent years. However, several important research gaps remain. Based on the literature review, the researchers identify two critical issues. First, the majority of ASAG models are trained and tested on responses to the same prompt which raises concerns about their robustness accross different prompts. Second, many existing approaches typically treat grading task as a binary classification problem. The research aims to bridge these gaps by developing an ASAG system that closely reflects real-world assessment scenarios through multiclass classification approach and cross-prompt evaluation. It is implemented by training the proposed models on 1,505 responses across 9 prompts and testing on 175 responses from 3 distinct prompts. The grading task is addressed using regression and classification techniques, including Linear Regression, Logistic Regression, Extreme Gradient Boosting (XgBoost), Adaptive Boosting (AdaBoost), and K-Nearest Neighbors (as a baseline). The grades are categorized into five classes that are represented by grade A to E. Both manual and algorithmic data augmentation techniques, including Syntactic Minority Oversampling Technique (SMOTE), are employed to address class imbalance in the sample data. Across multiple testing scenarios, all five models demonstrate consistent performance, with Linear Regression outperforming others. During the validation process, it achieves a high accuracy of 0.93, indicating its ability to correctly classify the responses. In the testing phase, it achieves a weighted F1-Score of 0.79, a macro-averaged F1-Score of 0.75, and an RMSE of 0.45. The results suggest relatively low prediction error.**

***Index Terms*—Cross-Prompt, Automatic Short Answer Grading (ASAG), Prompt-Specific**

I. Introduction

AUTOMATIC Short Answer Grading (ASAG) is often mistakenly equated with Automatic Essay Scoring (AES), which pertains to another field of research area [1]. While they are closely related, they differ in several aspects. The differences lie not only in their algorithmic tasks, but also in the techniques applied which depend on the question types as well as the response length. The task of ASAG's algorithm is to assess short natural language responses to objective questions using computational methods [2]. It is typically by matching it with a related reference model [3]. Meanwhile, AES system has a task of applying automation algorithms to evaluate the quality of written essay responses without the intervention of a human grader [4]. The ASAG for question type is objective or closed-ended. It requires answers that describe facts and statements, typically ranging from a single phrase to a paragraph in length [2]. Conversely, AES for question is subjective. It demands responses that present opinion or reasoning, ranging from two paragraphs to several pages in length [1].

Research on ASAG has progressed significantly, and numerous solutions have been proposed. However, there are still research gaps that need to be addressed. Firstly, the majority of ASAG systems are developed under the assumption of grading responses to the same question (i.e., prompt) during the training process [5–7]. This setup is vastly different from real-world scenarios, where ASAG is expected to grade responses to entirely new prompts or questions that are not part of the training data. Secondly, ASAG systems often treat the grading task as a binary classification [5–9]. In other words, they assign a holistic score to responses, using two scoring categories, i.e., 0 for incorrect or unsatisfactory and 1 for correct or satisfactory. Again, the setting is far from the real education system, where scores are often given in the range of $R[0, 5]$, $R[0, 10]$, $R[0, 4]$ or even $R[0, 100]$. Thirdly, when ASAG is treated as a multi-class classification, ensuring a fair distribution of data samples across

Received: April 25, 2025; received in revised form: Aug. 20, 2025; accepted: Aug. 20, 2025; available online: Oct. 13, 2025.
*Corresponding Author

each class is crucial. However, this setup is rarely found when dataset is acquired from a real-world case, often resulting in imbalanced data samples. The imbalanced data may severely degrade the performance of a standard classification model. It means that the prior probabilities of different classes vary significantly [10], causing the classifier to make biased decisions or predictions that favor the majority class, the one with the highest number of samples [11].

The research aims to solve the aforementioned problems for an ASAG system with an Indonesian prompt-response dataset. The main challenge lies in how to design ASAG system that closely reflects the real-world scenario. To address this challenge, the researchers propose a two-tier ASAG system. The tasks of the first tier are to measure the similarity between responses or student answers (SA) and their reference or teacher answer (TA) and to extract some features representing both SA and TA. The second tier deals with the grading task by applying classification and regression approaches. The researchers propose a cross-prompt model to be applied during the training and testing phases. Meanwhile, to address the problem of imbalanced data, the researchers propose using repeated Syntactic Minority Oversampling Technique (SMOTE).

A. Related works

Previous research identifies three key dimensions of natural language question types which separate short answer question from fill-in-the-gap and essay questions, such as length, focus, and openness [2]. The length of short answer questions ranges from a phrase to a paragraph. The marking technique for short answer questions focuses on the content of the answer, which aligns student's responses with expert's model [2, 12]. This means that ASAG requires a reference answer to determine correctness or adequacy of responses [13]. Thus, a part of ASAG's tasks falls within textual similarity area which is applicable also to text retrieval [14]. In contrast, the grading techniques of AES systems focus on metrics that broadly correlate with writing style [2] and concepts [1].

For the third dimension concerning the openness of question, ASAG systems require answers to objective questions or close-ended one, expressing facts and statements [2], definitions of given terms [12, 15], or description on specific events or given circumstances [8]. An ASAG system should be also capable of accommodating the semi open-ended question that requires learners to list specific facts and express their subjective opinions based on a specific context" [9]. Both close-ended and semi-open ended questions requiring answers in natural language are regarded as a valid technique for assessing higher order learning process [1]. It is because their answers need to recall the external knowledge of learners [2].

Some ASAG systems are designed with two main modules [12, 16], but many consist of only a single module [7, 15, 17] to effectively grade student responses for the aforementioned question types. Most single-module ASAG systems directly use similarity scores to grade student responses [17], while two-module ASAG systems separate the scoring task from model training into distinct modules. Some ASAG systems construct textual features by applying BERT [8], Continuous Bag of Words (CBOW) [9], and Word2Vec [12] or by building semantic networks using word graphs [16]. The most popular similarity metric used is cosine similarity [7, 12, 15, 17, 18]. As for the scoring task, some ASAG systems use classification approaches, such as SVM [9, 16] or Naïve Bayes and Decision Tree [9], while others make use of regression models, such as Logistic Regression [9] and Bayesian Linear Ridge Regression [19]. A distinct method is proposed. It highlights explainability of ASAG outcomes by combining transformer-based classifier, Scientific Bidirectional Encoder Representations from Transformers (SciBERT), with explainable SHapley Additive exPlanations (SHAP) module or Integrated Gradients to generate language explainability for each prediction [20]. Meanwhile, Funayama et al. have used keyphrases that answers should contain to increase scores [21].

For grading techniques, most ASAG systems focus on prompt-specific scoring. The term "prompt", borrowed from the AES field, refers to questions presented in a test. They define a writing instruction or topic [22]. A prompt-specific model segments their dataset into n subsets based on the number of prompts [4]. It trains and tests an AES model on each subset, resulting in n-prompt specific models for n-question prompts [4, 5, 8]. In contrast, the cross-prompt models are trained on non-target prompts [22, 23]. For example, the prompt-response pairs are not included in the testing data. These models should be able to predict the holistic or overall scores for target-prompt responses [23].

II. Research Method

In this research, the researchers propose a two-tier ASAG system: Textual Similarity and Feature Engineering (TSFE) and a scoring module. From this point onward, the first module is referred to as textual similarity module or TSFE for short. The scoring task is performed through regression and classification approaches as shown in Fig. 1.

Figure 1 displays that the ASAG model receives three inputs which are prompts or questions: teacher

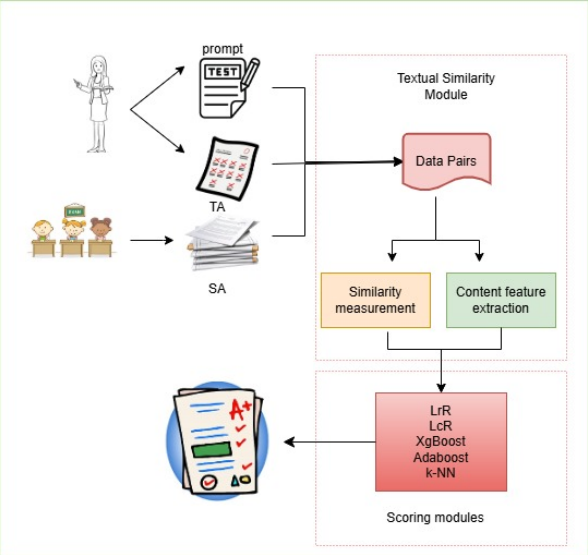

Fig. 1. The architecture of the proposed system. It has Linear Regression (LrR), Logistic Regression (LcR), Adaptive Boosting (AdaBoost), Extreme Gradient Boosting (XgBoost), and k-Nearest Neigbor (k-NN).

answer (TA), and student answer (SA). These inputs are structured into data pairs, wherein each prompt leads to a data identifier or ID. In this configuration, a single prompt functions as the key, with its associated values comprising of one teacher answer key (TA) and a corresponding set of student response (SA). The task of TSFE module is to extract the features through two steps, such as. comparing the textual similarity and the content features of the data pairs. The scoring module conducts the grading by applying some regression and classification models.

A. Data Acquisition

The data in the research are acquired manually by collecting and selecting pairs of questions and their respective responses from courses taught by researchers. Additionally, the data pairs are derived from the examinations conducted at a high school located in East Java, Indonesia. Some requirements for selecting pairs of questions and their respective responses as dataset are as follows:

1) The questions are the closed-ended or semi-open-ended type, with predetermined answers. However, the responses are expressed in natural language as opposed to a multiple choice or true-false questions,
2) The responses should reflect students' recall of their knowledge, particularly in relation to definition, explanation, or description on factual information or narration of events within specific contexts,
3) The question-response pairs should be available in a digital format, be evaluated, and include TA as a response reference,
4) The total number of responses per question should be more than 30, otherwise they are discarded,
5) The graded responses should have meta-information, such as grade range and maximum points.

TABLE I
THE CONVERTED RANGE OF GRADES AND LABELS.

Items	1	2	3	4	5
Ranges	0–4.0	4.1–5.49	5.5–7.0	7.1–8.49	8.5–10
Labels	E	D	C	B	A

Since the dataset is compiled from different courses and instructors, it exhibits variability in both range and distribution of grades, which affect the prediction performance of the classifiers. To address this issue, the grade ranges (0–4, 0–10, 0–12, and 0–100) are first normalized to a common scale of 0–10. This conversion is performed manually using equations in Microsoft Excel. Subsequently, the normalized grades are classified into five categories–A, B, C, D, and E–which serve as the corresponding data labels. This grading method reflects the actual system used in the department, where final grades are assigned using letters A–E, converted from numerical scores [24]. Table I shows the numerical grade ranges and their corresponding letter grades, which serve as class labels.

Next, two types of data augmentation are applied: the manual and the algorithmic ones. The manual data augmentation process aims to provide a minimum representation of five natural samples in each category, facilitating the effective implementation of algorithmic augmentation. This approach addresses the issue of data imbalance that certain classes either lack samples entirely or exhibit significant disparities in sample quantity as shown in Table II. To achieve this, three teaching assistants are employed to generate natural responses containing intentional errors corresponding to the grade categories with an insufficient number of samples. Given their prior experience with grading, they are well-qualified to replicate responses that align with the characteristics of various grading levels. To support the development of a cross-prompt ASAG model, the dataset is partitioned according to prompt identifiers (ID) and their corresponding TA-SA pairs. The prompts identified with ID01–ID03 are used as testing data, while prompts with ID04–ID12 serve as training data. This approach contrasts with prompt-

TABLE II
EXAMPLES OF DATASET STATISTICS BEFORE AND AFTER MANUAL AUGMENTATION.

Grade	Condition	ID04	ID05	...	ID12
A	Before	4	0	...	0
	After	5	5	...	5
B	Before	2	3	...	7
	After	5	5	...	7
C	Before	0	0	...	13
	After	5	5	...	15
D	Before	12	5	...	2
	After	12	5	...	5
E	Before	5	22	...	0
	After	5	22	...	5

An Example of dataset comprising 1 prompt and a set of TA and SAs

PID	Prompt	Teacher's Answer Keys (TA)	Student Answers (SA) / Responses
03	apakah yang disebut dengan "mounting" sebuah partisi atau volume?	memasangkan sebuah storage volume (partisi) ke sebuah node yang dapat diakses pada sebuah sistem komputer. Filesystem akan dibaca oleh sistem operasi untuk mengetahui informasi dan isi mengenai volume tersebut.	perintah yang digunakan untuk membuka sebuah device yang akan digunakan. a command used to access or initialize a device for use.
	what is meant by "mounting" a partition or volume?	Attaching a storage volume (partition) to a node so that it can be accessed by a computer system. The operating system reads the filesystem to obtain information about the volume and its contents.	proses mengkaitkan sebuah sistem berkas yang baru ditemukan pada sebuah piranti ke struktur direktori utama yang sedang dipakai. The process of attaching a newly detected filesystem on a device to the currently used main directory structure.
			mounting merupakan tindakan mengasosiasikan storage device (flashdisk,HDD,CDROM dll) ke lokasi tertentu pada directory tree linux (dibawah root directory /), hal ini perlu dilakukan karena Linux hanya mempunyai satu directory tree dengan induk atau root directory yang diberi simbol slash atau garis miring. Mounting is the act of associating a storage device (such as a flash drive, HDD, CD-ROM, etc.) with a specific location in the Linux directory tree (under the root directory /). This step is necessary because Linux has only one directory tree, with a single parent or root directory represented by a slash (/).

Fig. 2. An example of testing data.

specific ASAG models, which split the responses (SA) of a prompt into separate training and testing data. An example of testing data consisting a tripple of prompt-TA-SA is shown by Fig. 2.

B. Data Preprocessing

The dataset preprocessing is implemented through a method called $preprocessing()$ which defines text normalization processes such as case folding, stopword elimination, and others. Case folding is performed by applying $lower()$ function to convert all texts to lowercase, followed by punctuation removal using Regular Expression (RegEx). The next step is tokenization which utilizes $word\_tokenize()$ function from Natural Language ToolKit (NLTK) library. Like tokenization, the stopword removal is performed with Sastrawi stopwords which is provided by NLTK. Its stopword list comprises 758 distinct words. The normalized text, initially represented as a list, is then concatenated into a single string. Finally, the $preprocessing()$ method outputs two strings: normalized text without stopword and normalized text with stopwords.

C. The Textual Similarity and Feature Engineering (TSFE) Module

As mentioned earlier, the task of TSFE module is to measure content similarity between each TA and its corresponding SA. In addition, it extracts textual features: the Type-Token Ratio (TTR) and token counts. The similarity score is treated as one of the features for scoring, alongside the TTR and token count.

The similarity between each TA-SA pair is computed first using a custom-defined function, $tfidf\_weighting()$, which generates term vectors based on Term Frequency – Inverse Document Frequency (TF-IDF) weighting. In contrast to the conventional approach in the field of Information Retrieval (IR), where TF-IDF is computed across the

entire document corpus, the research computes TF-IDF values individually for each prompt–TA–SA pairs. Consequently, the TF-IDF calculation is performed iteratively for each prompt. This design choice is based on the assumption that scoring SAs within an ASAG system is independent of SAs associated with different prompts. Therefore, an SA's relevant corpus is the prompt-TA-SA pair in which it belongs. In other words, the semantic content of an SA is more closely related to its associated TA and the other SAs under the same prompt than to TA-SA pairs from different prompts.

Following the computation of TF-IDF vector, the similarity between TA and each SA is measured using Cosine Similarity, which is implemented through a custom-defined function, $get\_cossim()$. This function utilizes Cosine Similarity metric provided by the scikit-learn library. Its computation is based on the Eq. (1). To ensure consistency with the TF-IDF computation, the Cosine Similarity is calculated through a two-level iterative process. The first iteration corresponds to the number of distinct prompts, while the second iteration is performed for each SA within a given prompt. The ta refers to teacher's reference, and sa denotes the individual response or student answer. The V refers to the number of similar tokens in ta and sa, while i stands for the index of each similar token.

$$cos(\overrightarrow{ta}, \overrightarrow{sa}) = \frac{\sum_{i=1}^{|V|} \overrightarrow{ta_i} * \overrightarrow{sa_i}}{\sqrt{\sum_{i=1}^{|V|} \overrightarrow{ta_i^2}} \sqrt{\sum_{i=1}^{|V|} \overrightarrow{sa_i^2}}} \quad (1)$$

The TTR is calculated by dividing the number of unique words (type) by the total number of words (token). The TTR serves as an indicator of vocabulary richness, reflecting the extent to which variety of words are used rather than relying on the frequent repetition of the same words. The token count indicates the length of SA. All features along with the prompt ID, TA, SA, grade range, and their labels are then formulated in a data frame and saved in CSV file format, which constitutes the output of the TSFE module.

D. Scoring Module

The research perceives that the regression and ensemble classifiers can yield optimal predictive performance in scoring the SAs associated with the short answer prompts. For this reason, the research turns to Linear Regression, Logistic Regression, AdaBoost, XgBoost, and k-Nearest Neigbor. These five models are built in the following steps:

1) Importing the aforementioned models from $linear\_model$ package for Linear Regression and Logistic Regression, xgboost for XGBClassifier, ensemble package for AdaBoost, and neighbors fork-Nearest Neigbor,
2) Creating an object for each model,
3) Initializing each object as calling the training and testing functions.

Prior to model training, the features generated by the TFSE module are normalized using the MinMax scaler. Feature normalization is a crucial process, as the feature values vary significantly in scale. For instance, Cosine Similarity values range from 0 to 1, whereas token counts span from 10 to 200. With MinMaxScaler, the values of each feature are converted into a scale of 0.0–1.0.

Data augmentation is conducted in two distinct stages. The manual augmentation is carried out during the preprocessing phase, whereas algorithmic augmentation is applied immediately prior to model training. For the algorithmic data augmentation, the research applies SMOTE which has proven to be a robust method for addressing a noisy imbalanced data problem [10]. It is reported that SMOTE creates syntactic new samples of a minority class by using interpolation between minority class samples' neighborhood [24].

Unlike previous studies [10, 19], the research applies SMOTE repetitively for each class within the prompt-TA-SA pairs. It is conducted in two iterations. The innermost iteration is defined based on the number of classes corresponding to the five distinct grade categories. In contrast, the outer iteration is structured around the number of prompt IDs. The SMOTE process is applied exclusively to the training data, as SAs in the testing set come from different prompts. Prior to the application of SMOTE, the training set contains 642 SAs. This number increases to 1,505 following the oversampling process.

The model training is conducted using $fit()$ function. An exception is made for the AdaBoost model, which requires normalization of the target labels (Y_train). It is accomplished using the $LabelEncoder()$ function from the preprocessing module of the scikit-learn library. To support a cross-prompt ASAG system, each model is sequentially trained on every prompt-TA-SA pair enabling a single model to learn across n different prompts. In contrast, the prompt-specific ASAG system trains each prompt separately, resulting in n-models. The n denotes the number of prompts in the training data. Figure 3 displays the process of training and validating the model.

E. Evaluation Matrices

To assess the performance of the proposed system, three matrices are employed: Accuracy, F1-Score,

Algorithm 1 Data Augmentation and Training

Require: $prompts, TA, SAfeat, grades$
Ensure: $df = pd.DataFrame(prompts, TA, SAfeat, grades)$
  for $prompts = 1$ to N **do**
    $dataPerPrompt \leftarrow df.query('promptID')$
    $augmentedData \leftarrow SMOTE(dataPerPrompt)$
    $trainedPrompt \leftarrow model.TRAIN(augmentedData)$
    $valScore \leftarrow stratifiedkfold(trainedPrompt)$
  end for

Fig. 3. Algorithm for data augmentation and training.

and Root Mean Square Error (RMSE). Accuracy is used during the validation process, whereas F1-Score and RMSE are applied in the testing phase. These three metrices are implemented by importing the $classification\_report()$ function provided by scikit-learn library. Based on the confusion matrix, scikit-learn computes F1 as shown by Eq. (2). Then, the Macro-Averaged F1 (MAF) and Weighted-Averaged F1 (WAF) scores for multiclass classification are shown in Eqs. (3) and (4), respectively. TP refers to True Positive, FP denotes False Positive, and FN corresponds to False Negative. The N refers to the total number of classes, while Sp_i indicates the support proportion of classifier (support values). F1 metric is a widely used performance measure that represents the harmonic means of Precision and Recall that is also known as Sensitivity. MAF computes the mean of the F1 scores across all classes, treating each class equally as displayed in Eq. 3. In contrast, WAF takes into account the number of instances in each class, represented by Sp_i, making it well suited for evaluating the performance of multi-class classifiers, especially when class distributions are imbalanced. It explains why F1 score is multiplied to Sp_i in Eq. (4).

$$F1 = \frac{2 * TP}{2 * TP + FP + FN}, \tag{2}$$

$$MAF = \frac{\sum F1_i}{N}, \tag{3}$$

$$WAF = \sum_{i=1}^{N} F1_i * Sp_i. \tag{4}$$

The RMSE is employed due to its straightforward interpretability as a measure of prediction error. It is computed by taking the square root of the average of the squared differences between predicted and actual values as shown in Eq. 5.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}i\|}{N}}. \tag{5}$$

Here, N denotes the number of SAs, and $y(i)$ is actual score of SA, and $\hat{y}$ is the predicted score of SA. RMSE is widely used and preferable in regression models due to its simple interpretability [25]. Another advantage of RMSE is that it is expressed in the same units and scale as the original data [25]. It allows for more direct and meaningful comparisons between predicted and observed values. It makes RMSE especially useful when evaluating model performance in real-world applications.

TABLE III
THE TRAINING DATASET BEFORE AND AFTER AUGMENTATION USING SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE (SMOTE).

PromptIDs	# Before	# After
IDPSJ04	32	60
IDPSJ05	42	110
IDPSJ06	44	110
IDPSJ07	40	65
IDPSJ08	112	315
IDPSJ09	103	240
IDPSJ10	114	205
IDPSJ11	120	325
IDPSJ12	35	75

III. RESULTS AND DISCUSSION

To reduce the prediction bias, the manually augmented dataset presented in Table II is further enhanced through algorithmic augmentation using SMOTE technique. This approach facilitates a more balanced distribution of samples across all classes. Table III summarizes the training dataset statistics per prompt. Prior to SMOTE, the number of SAs per prompt ranges from 32 to 120. After SMOTE, this range increases to 60 to 325. The training data comprises 9 prompts, totaling in 1,505 SAs.

Since SMOTE generates synthetic samples in the form of feature vectors, it is applied after preprocessing and feature extraction, but before model training. Following a cross-prompt ASAG setup, SMOTE is applied independently to each prompt. It means that within each prompt, SMOTE is operated on samples of each class, generating new samples based on nearest neighbors within that class. For example, it considers the prompt identified by the code IDPSJ04, as presented in Table III. Before applying SMOTE, this prompt has a total of 32 data samples, indicating an imbalanced distribution of samples across the five classes. However, the total number of samples increased to 60 after SMOTE (12 samples $\times$ 5 classes), ensuring equal representation across all classes.

A. Validation Process

To evaluate the training process, a stratified 2-fold cross-validation technique is implemented with

a ratio of 70% to 30% training-testing split. Accuracy is used as the evaluation metric. The SA for validation is derived from the same prompts as in training. During validation process, various SMOTE variants –Adaptive Synthetic Sampling (ADASYN), SMOTETomek, SVMSMOTE, and KMeans-SMOTE– are tested. However, they perform significantly worse compared to basic SMOTE. Consequently, only basic SMOTE is applied for augmenting data.

Figure 4 illustrates the performance of various models in terms of validation accuracy. Notably, Linear Regression achieves the highest validation accuracy of 0.93 when trained without applying SMOTE. The result indicates that it performs very well even with the imbalanced data. However, after applying SMOTE, the validation accuracy of Linear Regression slightly decreases to 0.92. This decrease is minor and statistically insignificant, suggesting that SMOTE does not substantially impact performance of Linear Regression positively or negatively. It also implies that Linear Regression may be relatively robust to class imbalance in this particular setting.

Similar to Linear Regression model, the prediction performance of AdaBoost model during validation is not significantly affected by the application of SMOTE. Its validation accuracy decreases by only 0.02, indicating a slight and negligible decline. The XGBoost model, which shows the lowest performance among the tested models, is similarly unaffected by SMOTE. It indicates that oversampling has minimal impact on its validation accuracy

Unlike Linear Regression, AdaBoost, and XGBoost, SMOTE significantly improves the performance of k-Nearest Neighbor and Logistic Regression. Without SMOTE, k-Nearest Neighbor and Logistic Regression achieve validation accuracies of 0.65 and 0.60, respectively. After applying SMOTE, their accuracies rise notably to 0.79 for k-Nearest Neighbor and 0.73 for Logistic Regression, highlighting their sensitivity to class imbalance.

B. Testing Process

To support cross-prompt ASAG, the testing data, comprising SAs, are derived from entirely different prompts than those used during training, thereby reflecting real-world scenarios where input prompts and corresponding responses typically differ from the training data. Accordingly, the research selects three prompts from the subjects of Pendidikan Pancasila dan Kewarganegaraan (PPKN), Operating Systems, and Digital Humanities. The testing set comprises a total of 175 SAs. The statistics of the testing data are presented in Table IV. Combined with the training data, the total number of dataset consists of 1650 SAs.

TABLE IV
THE STATISTICS OF THE TESTING DATA.

PromptIDs	Subjects	Question Category	#Student Answers
IDPSJ01	PPKN	Descriptive explanatory	101
IDPSJ02	Operating System	Explanatory definition	41
IDPSJ03	Digital Humanities	semi-open-ended, explanatory,and giving examples	33

As shown in Table IV, the SAs in the testing data include both closed-ended and semi-open-ended questions. A closed-ended example asks for the definition and elements of Digital Humanities, where the expected answer closely matches the TA. In contrast, a semi-open-ended question asks for examples of Digital Humanities projects and their URLs, leading to more varied SAs. It poses a greater challenge for the ASAG system to predict them accurately.

Unlike in validation, the model's performances are assessed using the F1-Score and RMSE. The F1-Score captures the balance between FP and FN, while RMSE quantifies the average magnitude of prediction errors. For F1-Score, the research computes MAF and WAF. These metrics are chosen to examine classification accuracy as well as error magnitude. Observing consistency in prediction tendencies across metrics indicates the model's reliability, regardless of the specific values yielded by each metric. The experiment results using MAF and WAF are presented in Table V. As shown in Table V, Linear Regression achieves the highest F1-Score with 0.75 for MAF and 0.79 for WAF. However, these scores decline to 0.60 and 0.75, respectively, after the application of SMOTE. Then, XGBoost demonstrates the poorest performance, with F1-Scores below 0.1 for both MAF and WAF. In contrast, SMOTE notably improves the performance of Logistic Regression and k-Nearest Neighbors (k=3). The F1-Score for the AdaBoost model remains unchanged with SMOTE. These F1-Scores show a strong correlation with the validation accuracy rates, suggesting that the models not only perform well in the overall accuracy in validation process but also maintain a good balance between precision and recall during the testing. This alignment indicates that the models deliver reliable and consistent predictions, particularly in handling both correct classifications and class imbalances effectively.

RMSE scores are observed with and without the application of SMOTE to evaluate the impact of class balancing on prediction error. Unlike MAF and WAF scores, which are normalized between 0 and 1, RMSE values can range from 0 to infinity (∞), depending on

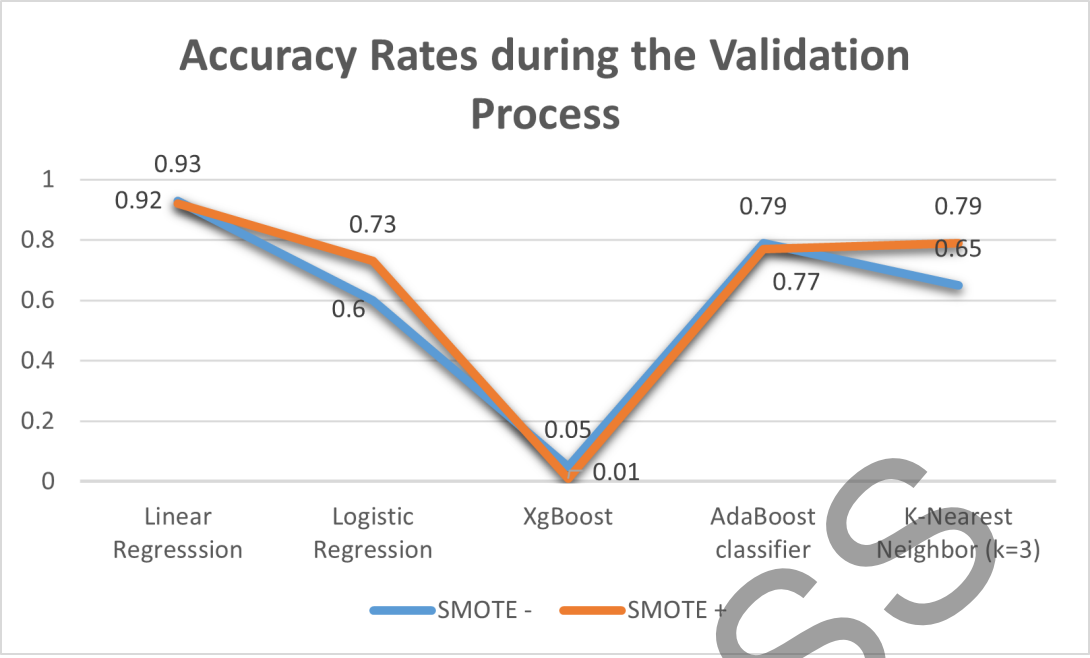

Fig. 4. The accuracy of model predictions during validation process.

TABLE V
MACRO-AVERAGED & WEIGHTED-AVERAGED F1 SCORES OF 5 MODELS

Models	Without SMOTE		With SMOTE	
	MAF	WAF	MAF	WAF
Linear Regression	**0.75**	**0.79**	**0.6**	**0.75**
Logistic Regression	0.18	0.18	0.5	0.58
Extreme Gradient Boosting (XgBoost)	0.06	0.04	0.07	0.05
Adaptive Boosting (AdaBoost)	0.59	0.65	0.59	0.65
k-Nearest Neigbor	0.25	0.24	0.57	0.61

the magnitude and scale of the prediction errors [25]. A lower RMSE indicates that the model's predictions are closer to the actual values, with 0 representing perfect accuracy. Conversely, higher RMSE values reflect larger disparity between predicted and actual outcomes, signalling poorer model performance. Figure 5 presents the RMSE values of five models in predicting SA grades, comparing results with and without SMOTE.

Figure 5 shows that the best RMSE value is achieved by Linear Regression with 0.45, followed by AdaBoost (0.52). Logistic Regression and k-Nearest Neigbor (k=3) yield RMSEs of 1.41 and 1.31, respectively. Meanwhile, XGBoost exhibits the worst performance by RMSE at 1.89. Using predictive error of k-Nearest Neigbor as a baseline, only Logistic Regression and XGBoost perform worse, while Linear Regression and AdaBoost demonstrate improved predictive accuracy. These RMSE values represent the models' predictive errors without the use of SMOTE. In contrast, applying SMOTE leads to a reduction in RMSE and improved predictive accuracy for three models: Logistic Regression, XGBoost, and k-Nearest Neighbor. Specifically, their RMSE scores drop to 0.87, 1.53, and 0.82, respectively. The results indicate a notable improvement in prediction quality due to class balancing. Meanwhile, the AdaBoost model shows no significant change in its RMSE score. It suggests that its performance remains largely unaffected by SMOTE.

Though RMSE values range from 0 to ∞ [26], their effective upper bound can be inferred from the maximum values in the observed testing data. In the dataset, the normalized grades span from 0.00 to 10.00 and are classified into five categories (Table I), with an inter-category difference of approximately 1.5 points. Considering this difference point in conjunction with the accuracy, MAF and WAF scores, the RMSE values obtained from the experiments can be interpreted accordingly as follows:

1) RMSE values $\leq$ 0.59 indicate high predictive accuracy, effectively assigning predicted grades

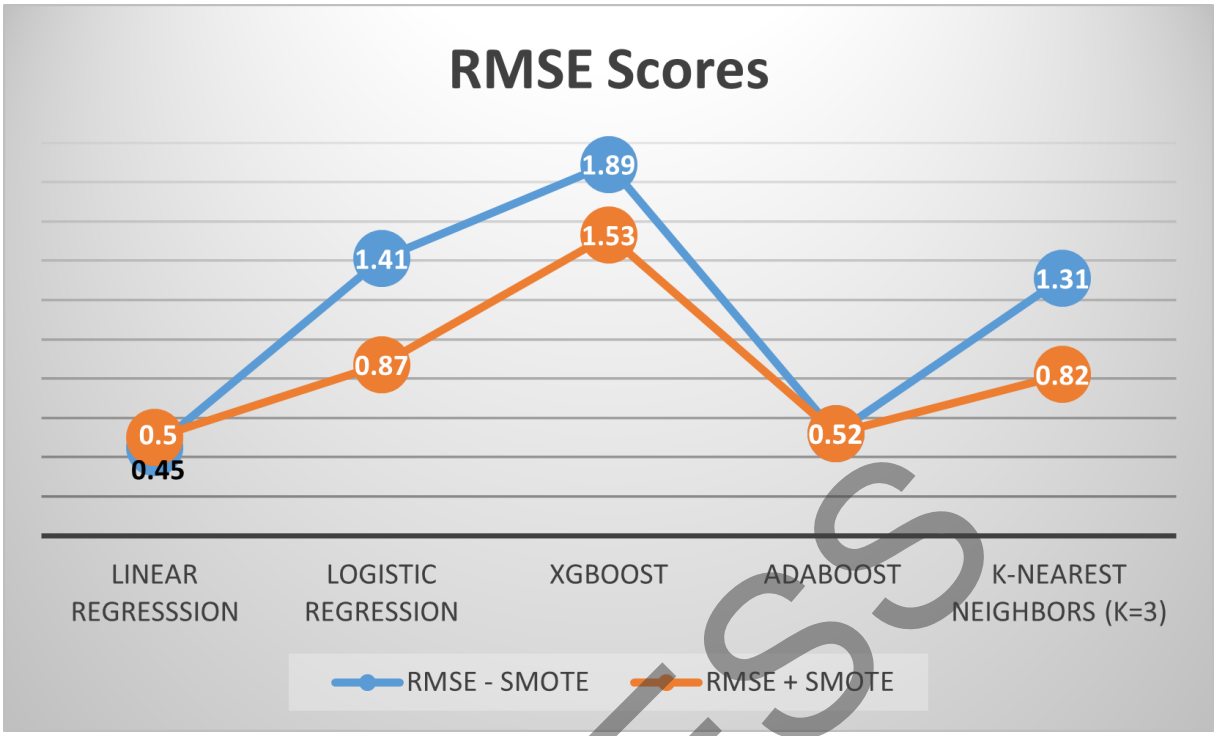

Fig. 5. The Root Mean Square Error (RMSE) scores of five proposed models.

within the same categorical grade range as the gold label,

2) An RMSE between 0.6 to 1.49 is considered acceptable as it is accompanied by satisfactory level of accuracy, MAF, and WAF as well,
3) RMSE $\geq$ 1.5 indicates a high degree of predictive inaccuracy, suggesting that the model's estimations deviate substantially from the actual grade labels.

Analysis of the validation and assessment results indicates that the evaluated models exhibit consistent performance across all applied metrics and testing scenarios, with Linear Regression consistently achieving the highest predictive accuracy (Fig. 4 and Table V). The Linear Regression model demonstrates strong predictive performance, even without the need for balanced data, highlighting its robustness to class imbalance. This result is evidenced by its high accuracy score of 0.93, along with a MAF of 0.79, a WAF of 0.85, and a low RMSE of 0.45. These metrics collectively indicate that Linear Regression consistently delivers accurate and reliable predictions across all classes, making it one of the best-performing models in the evaluation.

The XGBoost model demonstrates the weakest performance across all testing scenarios, with an RMSE of 1.89. However, the application of SMOTE leads to a noticeable improvement in its predictive performance. In contrast, applying SMOTE results in a negligible decline in the predictive performance of Linear Regresion model. Overall, SMOTE tends to enhance the predictive performance of models, such as AdaBoost, Logistic Regression, and k-Nearest Neighbor. However, the performance of AdaBoost remains unaffected by SMOTE.

The notably higher accuracy rates observed during the validation process, compared to the corresponding MAF and WAF scores during the testing phase, are justifiable. This discrepancy can be attributed, in part, to the overlapping between the validation and training data, as SAs for validation process are derived from the same prompts whose responses are partially included in the training set. Furthermore, the testing phase applies a cross-prompt approach, wherein the predicted grades of SA (responses) originate from prompts not encountered during training. As a result, the observed MAF, WAF, and RMSE scores are more satisfactory rather than very high. This result highlights the key challenge inherent to cross-prompt ASAG.

IV. Conclusion

This study presents a two-tier cross-prompt ASAG system designed to closely mirror real-world assessment scenarios. The first tier involves TSFE, wherein

the similarity between a prompt and SA is computed and used as a feature, alongside TTR and SA length. The second tier performs scoring using a classification-based approach, experimenting with two regression and three classification models. The SMOTE is applied to address class imbalance.

The performance of the proposed system is evaluated using MAF, WAF, and RMSE, with and without SMOTE application. Across multiple scenarios, all five models demonstrate consistent performance, with Linear Regression achieving the highest scores: 0.79 for WAF, 0.75 for MAF, and 0.45 for RMSE. These scores are notably lower than accuracy score of Linear Regression during validation, which reaches 0.93. This discrepancy arises because the validation process is conducted using a prompt-specific design – scoring SAs from known and trained prompts. In contrast, the testing scenario of the research evaluates the proposed system with entirely unknown prompts and their corresponding responses (SA) to support a cross-prompt ASAG system.

Based on evaluation results, the predictive performance of the proposed system can be considered moderate. This outcome is partly attributable to inherent challenges associated with cross-prompt ASAG, some of which remain unaddressed in the research. Notably, the correlation between a prompt and its corresponding SA has not yet been explored. Future research can focus on addressing these limitations particularly by modelling prompt-response relationship and incorporating semantic similarity measures between TA and SA. Future research can focus on addressing these limitations particularly by developing models that explicitly capture the relationship between teacher's prompt and student's responses. Additionally, incorporating semantic similarity measures between them may lead to increase predictive accuracy, especially in cases where the teacher's prompts are not present in the training dataset, which is often the case in real-world scenarios.

Acknowledgement

The research was supported by a grant from Institute of Research and Community Service (LPPM), Universitas Kristen Duta Wacana and the joint research scheme of Multimedia University, Malaysia. The authors are indebted to both institutions which provided a grant to assist with the research.

Author Contribution

Conceived and designed the analysis, L. D. K.; Contributed to methodology and coding, L. D. K.; Wrote the paper, L. D. K.; Collected the data, A. W. M.; Performed the analysis, A. W. M. and H. S. C.; and Proofread the article submitted on the first round, H. S. C.

Data Availability

The data that support the findings of the research are available from the corresponding author, Lucia D. Krisnawati, upon reasonable request. The dataset is currently expanded and restructured. Once finalized, it will be made publicly available in a designated database.

References

[1] D. Ifenthaler, "Handbook of open, distance and digital education." Singapore: Springer, 2022, ch. Automated essay scoring system, pp. 1057–1071.

[2] S. Burrows, I. Gurevych, and B. Stein, "The eras and trends of automatic short answer grading," *International Journal of Artificial Intelligence in Education*, vol. 25, no. 1, pp. 60–117, 2015.

[3] A. Sahu and P. K. Bhowmick, "Feature engineering and ensemble-based approach for improving automatic short-answer grading performance," *IEEE Transactions on Learning Technologies*, vol. 13, no. 1, pp. 77–90, 2019.

[4] B. Cho, Y. Jang, and J. Yoon, "Rubric-specific approach to automated essay scoring with augmentation training," 2023. [Online]. Available: https://arxiv.org/abs/2309.02740

[5] E. Del Gobbo, A. Guarino, B. Cafarelli, and L. Grilli, "GradeAid: A framework for automatic short answers grading in educational contexts—Design, implementation and evaluation," *Knowledge and Information Systems*, vol. 65, no. 10, pp. 4295–4334, 2023.

[6] A. A. Septiandri, Y. A. Winatmoko, and I. F. Putra, "Knowing right from wrong: Should we use more complex models for automatic short-answer scoring in Bahasa Indonesia?" in *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*. Online: Association for Computational Linguistics, November 2020, pp. 1–7.

[7] M. R. R. Susanto, H. Thamrin, and N. A. Verdikha, "Performance of text similarity algorithms for essay answer scoring in online examinations," *Jurnal Teknik Informatika (JUTIF)*, vol. 4, no. 6, pp. 1515–1521, 2023.

[8] R. A. Rajagede, "Improving automatic essay scoring for indonesian language using simpler model and richer feature," *Kinetik: Game technology,*

Information System, Computer Network, Computing, Electronics, and Control, vol. 6, no. 1, pp. 11–18, 2021.

[9] L. Zhang, Y. Huang, X. Yang, S. Yu, and F. Zhuang, "An automatic short-answer grading model for semi-open-ended questions," *Interactive Learning Environments*, vol. 30, no. 1, pp. 177–190, 2022.

[10] J. Liu, "Importance-SMOTE: A synthetic minority oversampling method for noisy imbalanced data," *Soft Computing*, vol. 26, no. 3, pp. 1141–1163, 2022.

[11] N. U. Niaz, K. M. N. Shahariar, and M. J. A. Patwary, "Class imbalance problems in machine learning: A review of methods and future challenges," in *Proceedings of the 2nd International Conference on Computing Advancements*, 2022, pp. 485–490.

[12] F. F. Lubis, A. Putri, D. Waskita, T. Sulistyaningtyas, A. A. Arman, and Y. Rosmansyah, "Automated short-answer grading using semantic similarity based on word embedding," *International Journal of Technology*, vol. 12, no. 3, pp. 571–581, 2021.

[13] M. Chen and Y. Dong, "Design of exercise grading system based on text similarity computing," *Mobile Information Systems*, vol. 2022, pp. 1–7, 2022.

[14] L. D. Krisnawati, A. W. Mahastama, S. C. Haw, K. W. Ng, and P. Naveen, "Indonesian-English textual similarity detection using Universal Sentence Encoder (USE) and Facebook AI Similarity Search (FAISS)," *CommIT (Communication and Information Technology) Journal*, vol. 18, no. 2, pp. 183–195, 2024.

[15] U. Hasanah, T. Astuti, R. Wahyudi, Z. Rifai, and R. A. Pambudi, "An experimental study of text preprocessing techniques for automatic short answer grading in Indonesian," in *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*. Yogyakarta, Indonesia: IEEE, Nov. 13–14, 2018, pp. 230–234.

[16] N. H. Hameed and A. T. Sadiq, "Automatic short answer grading system based on semantic networks and support vector machine," *Iraqi Journal of Science*, vol. 64, no. 11, pp. 6025–6040, 2023.

[17] D. Wilianto and A. S. Girsang, "Automatic short answer grading on high school's e-learning using semantic similarity methods," *TEM Journal*, vol. 12, no. 1, pp. 297–302, 2023.

[18] F. Li, X. Xi, Z. Cui, D. Li, and W. Zeng, "Automatic essay scoring method based on multi-scale features," *Applied Sciences*, vol. 13, no. 11, pp. 1–18, 2023.

[19] J. S. Tan, I. K. T. Tan, L. K. Soon, and H. F. Ong, "Improved automated essay scoring using Gaussian multi-class SMOTE for dataset sampling," in *Proceedings of the 15th International Conference on Educational Data Mining*, England, UK, July 24–27, 2022.

[20] M. Tornqvist, M. Mahamud, E. M. Guzman, and A. Farazouli, "ExASAG: Explainable framework for automatic short answer grading," in *Proceedings of the 18th workshop on innovative use of NLP for building educational applications (BEA 2023)*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 361–371.

[21] H. Funayama, Y. Asazuma, Y. Matsubayashi, T. Mizumoto, and K. Inui, "Reducing the cost: Cross-prompt pre-finetuning for short answer scoring," in *International Conference on Artificial Intelligence in Education*. Tokyo, Japan: Springer, July 3–7, 2023, pp. 78–89.

[22] H. Do, Y. Kim, and G. G. Lee, "Prompt- and trait relation-aware cross-prompt essay trait scoring," 2023. [Online]. Available: https://arxiv.org/abs/2305.16826

[23] R. Ridley, L. He, X. Y. Dai, S. Huang, and J. Chen, "Automated cross-prompt scoring of essay traits," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 15. Virtual: Association for the Advancement of Artificial Intelligence, Feb. 2–9, 2021, pp. 13 745–13 753.

[24] Tim Kurikulum, *Panduan akademik kurikulum 2021 revisi 2023*. Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana, 2023.

[25] T. O. Hodson, "Root Mean Square Error (RMSE) or Mean Absolute Error (MAE): When to use them or not," *Geoscientific Model Development Discussions*, vol. 2022, pp. 1–10, 2022.

[26] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, 2021.