Enhancing Neural Automated Essay Scoring Accuracy by Removing Noisy Data Through Data Valuation

Takumi Shibata[1,*], Yuto Tomikawa[1], Yuki Ito[1] and Masaki Uto[1,*]

[1]*The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo, Japan*

Abstract

Essay examinations are widely used to assess expressive ability and logical thinking skills, but grading by humans incurs significant costs and presents challenges in ensuring reliability. To address these issues, automated essay scoring (AES) using artificial intelligence technologies has gained considerable attention. Although deep neural network-based AES methods have demonstrated high accuracy in recent years, constructing highly performing models requires training on large-scale, high-quality datasets of graded essays. In addition, rater biases and grading errors can introduce noisy data, which may negatively impact model performance. To address this, machine learning-based methods for detecting noisy data by estimating the value of each training sample have been proposed. In this study, we propose a method for constructing an AES model that leverages data valuation techniques to exclude noisy data. Furthermore, we integrate our method with a large language model-based data augmentation approach, enabling the selection of high-quality augmented data to further enhance AES accuracy in low-resource scenarios.

Keywords

Automated essay scoring, deep neural networks, data valuation, noisy data, large language models

1. Introduction

In recent years, there has been a growing emphasis on not only knowledge retention but also the development of practical skills such as expressive ability and logical thinking skills in education. Essay examinations have received significant attention as an approach to evaluating these higher-order abilities [1, 2, 3]. However, large-scale essay examinations incur substantial cost for manual grading due to the high volume of essays. Furthermore, factors such as rater biases and grading errors pose challenges in ensuring fairness and reliability in scoring [4].

To address these issues, automated essay scoring (AES) using artificial intelligence technologies has gained increasing attention in recent years [5, 6]. In particular, deep neural network-based AES models have been widely proposed and have demonstrated high accuracy [7, 8]. These neural AES models process essay texts as sequences of words and automatically extract latent features for score prediction within the model. A representative approach in neural AES is based on bidirectional encoder representations from transformers (BERT) [9], and numerous extended models have been developed [10, 11].

Although these neural AES models have achieved highly accurate scoring, the construction of highly performing models requires large-scale, high-quality datasets of scored essays. However, creating such high-quality datasets is not a trivial task. As mentioned earlier, factors such as rater biases and grading errors can lead to incorrect score assignments during the manual grading process, and these erroneous scores introduce noise into the training data, potentially degrading the performance of AES models. Therefore, a method for identifying and removing such noisy data during model training is highly desirable.

To address these challenges, machine learning research has actively explored techniques for filtering noisy data through data valuation, which estimates the value of each training data sample [12, 13, 14, 15, 16]. Data value is a concept that quantifies the contribution of each training sample to model

EvalLAC'25: 2nd Workshop on Automatic Evaluation of Learning and Assessment Content, July 26, 2025, Palermo, Italy
*Corresponding author.
shibata@ai.lab.uec.ac.jp (T. Shibata); uto@ai.lab.uec.ac.jp (M. Uto)
0000-0002-9330-5158 (M. Uto)

performance. In data valuation, samples that positively influence model performance are assigned higher values, while those that negatively impact performance receive lower values.

The simplest method for estimating data value is leave-one-out (LOO) [12], which evaluates the impact of removing a single data sample on model performance. Additionally, data Shapley [14], inspired by game theory, is another widely used data valuation method. More recently, a reinforcement learning-based approach called data valuation using reinforcement learning (DVRL) [15] has demonstrated both high accuracy and computational efficiency of data valuation. While these techniques have been employed in various machine learning domains, their use in AES remains highly limited.

Therefore, in this study we investigate a method that applies three representative data valuation methods, namely, LOO, data Shapley, and DVRL, to detect and remove noisy samples for training neural AES models. Through experiments using a benchmark dataset, we demonstrate that our method can identify noisy training samples, thereby improving score prediction accuracy.

In addition, we explore integration of our proposed method with data augmentation techniques, which are widely used in machine learning tasks, including AES. While data augmentation is often employed to increase the number of training samples, particularly in low-resource scenarios [17, 18, 19, 20, 21, 22, 23, 24], the quality of augmented data is not always guaranteed and may introduce noise. Therefore, we hypothesize that applying our method can effectively filter out noisy augmented data, leading to improved scoring accuracy. In our experiments, we apply our method to a data augmentation approach based on a large language model (LLM) [19, 24] and demonstrate its effectiveness.

2. Automated Essay Scoring

AES has evolved rapidly in recent years, driven by advancements in artificial intelligence [5, 6, 25, 26]. This section provides an overview of AES tasks and conventional AES methods, followed by an introduction to the AES model used in the present study.

2.1. AES Tasks

AES tasks can be categorized based on their scoring scope and prompt dependency. In terms of scoring scope, AES is broadly divided into *summative* and *analytical* scoring [5, 27, 28, 29, 30, 31]. Summative scoring, also called *holistic scoring*, refers to single-aspect scoring, where an overall score is assigned to an essay. In contrast, analytical scoring, also known as *trait scoring*, evaluates multiple aspects of an essay, such as content, organization, language use, and mechanics, assigning separate scores to each dimension.

Another key classification of AES tasks is based on prompt dependency, where prompts refer to specific essay assignments [5, 32, 33, 34, 35]. *Prompt-specific AES* is the task of constructing scoring models that are trained on essays from a single prompt to predict scores for essays written in response to the same prompt. In contrast, *cross-prompt AES* is the task of achieving generalized scoring across different prompts by utilizing training data from multiple prompts to build a model capable of scoring essays written in response to unseen prompts.

In this study, we focus on the most fundamental AES task: *prompt-specific* and *summative* scoring. This setting ensures that we train and test a single-aspect scoring model on essays responding to the same prompt.

2.2. Conventional AES Methods

Conventional AES methods under prompt-specific summative scoring settings have relied on feature-engineering approaches, which use manually designed features, such as essay length and the number of grammatical errors, to predict scores through regression or classification models [36, 37, 38]. While these methods offer interpretability, they require significant effort in feature selection and engineering, often necessitating customization for each specific essay prompt.

To address this limitation, deep neural network-based approaches have become increasingly popular. These methods automatically extract relevant features from data, eliminating the need for manual feature design. Early neural AES models have primarily utilized convolutional neural networks or recurrent neural networks [39, 40, 41, 42, 43, 44]. More recent studies have focused on leveraging pretrained transformer-based models [45], particularly BERT [9], which have demonstrated high performance and accuracy in AES tasks [11, 46, 47]. BERT and its variants undergo extensive pretraining on large-scale corpora, and fine-tuning on relatively small scored datasets has allowed them to achieve high accuracy in AES tasks.

Recently, LLMs have emerged as the next frontier in AES research. A key advantage of LLMs is their ability to perform various natural language processing (NLP) tasks, including AES, by processing task instructions with optional few-shot examples [48]. LLMs are highly adaptable across different NLP tasks while reducing the dependence on large labeled datasets for each specific task. Recent studies have explored the use of LLMs for AES, demonstrating that they can achieve reasonable scoring performance [26, 49, 50, 51, 52].

Although LLM-based AES has demonstrated acceptable accuracy under certain conditions, these models generally underperform compared with fine-tuned scoring models based on pretrained transformers [49, 50, 52, 53]. Given these trends, this study focuses on a fine-tuned BERT-based AES model as the foundation for our research.

2.3. BERT-based AES Model

BERT-based AES models are typically designed by incorporating a linear output layer with a sigmoid activation function into the BERT architecture. Specifically, let $\mathbf{z}$ denote the output vector of BERT corresponding to the [CLS] token, which is added at the beginning of each essay. The predicted score $\hat{y}$ is then computed through the following output layer:

$$\hat{y} = \sigma(\mathbf{W}\mathbf{z} + b) \tag{1}$$

where $\mathbf{W}$ and b represent the weight matrix and bias parameter, respectively, and $\sigma()$ denotes the sigmoid function. Fine-tuning is conducted using backpropagation with the following mean squared error (MSE) loss function.

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2 \tag{2}$$

where y_n and $\hat{y}_n$ denote the gold-standard and model-predicted scores for the n-th training sample, respectively, and N represents the total number of samples. Note that, because sigmoid activation is applied to the output layer, gold-standard scores y_n must be standardized into the range [0,1] during model training. Furthermore, during the inference stage, predicted scores must be linearly transformed back to the original score scale.

Although such neural AES models have achieved high accuracy, their performance can degrade when training datasets contain noisy data. To address this issue, this study proposes a method for filtering out potentially noisy samples from the training dataset by leveraging data valuation techniques.

3. Data Valuation

This section provides an overview of the three representative data valuation methods used in this study, namely, LOO, data Shapley, and DVRL. These methods assume that the training dataset $\mathcal{D}^t$ and validation dataset $\mathcal{D}^v$ are given, both of which consist of labeled samples (x, y), where x represents the explanatory variable and y denotes the corresponding label. Data values are assigned to the training samples in $\mathcal{D}^t$ based on their contribution to the performance of a predictor on the validation dataset $\mathcal{D}^v$.

3.1. Leave-One-Out

LOO is the simplest method for data valuation [12]. In this approach, the value of a specific training sample is determined by measuring the difference in the performance of a predictor when trained with and without that particular sample. More specifically, let $V(\mathcal{D}^t)$ denote the performance of a predictor trained on the training dataset $\mathcal{D}^t$ when evaluated on the validation dataset $\mathcal{D}^v$. The value of the n-th training sample is then computed as follows:

$$\phi_n = V(\mathcal{D}^t) - V(\mathcal{D}^t - \{n\}) \tag{3}$$

where $\mathcal{D}^t - \{n\}$ refers to the set obtained by removing the n-th sample from $\mathcal{D}^t$.

A known limitation of LOO is that if a removed sample was originally beneficial but the remaining training samples still enable the predictor to achieve sufficient performance, its value may be underestimated. Consequently, accurately estimating the value of individual training samples in large datasets remains challenging. To address this issue, the data Shapley method [14] was proposed.

3.2. Data Shapley

The data Shapley method applies Shapley values [54, 55] from game theory to estimate data values. In data Shapley, the power set of the training dataset is considered. The predictor is then trained and evaluated on all possible subsets to compute the value of each training sample. Letting $\mathcal{S}$ be any subset of $\mathcal{D}^t - \{n\}$, the value of the n-th training sample is computed as follows:

$$\phi_n = C \sum_{\mathcal{S} \subseteq \mathcal{D}^t - \{n\}} \frac{V(\mathcal{S} \cup \{n\}) - V(\mathcal{S})}{\binom{|\mathcal{D}^t|-1}{|\mathcal{S}|}} \tag{4}$$

where $\mathcal{S} \cup \{n\}$ represents the set obtained by adding the n-th sample of $\mathcal{D}^t$ to $\mathcal{S}$, and C is a scaling constant. Unlike LOO, the data Shapley accounts for contributions across all subsets, leading to more precise data valuation. However, since the computation requires evaluating 2^{N-1} combinations for a dataset of size N, the computational cost increases exponentially. To mitigate this issue, an approximation method called gradient Shapley (GS) has been proposed, which employs Monte Carlo sampling and gradient-based optimization. See [14] for the detailed algorithm.

Although GS reduces computational cost compared with the original data Shapley method, it remains computationally expensive. DVRL was therefore proposed to further enhance efficiency while maintaining high accuracy of data valuation [15].

3.3. Data Valuation Using Reinforcement Learning

DVRL consists of two components, namely, a predictor and a data value estimator, which are jointly optimized within a reinforcement learning framework. The DVRL training process is formulated as the following optimization problem:

$$\max_{\phi} \mathbb{E}_{(x,y)\sim\mathcal{D}^v}[R_\phi(\theta^*)] \;\text{ s.t. }\; \theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}^t}[h_\phi(x,y) \cdot \mathcal{L}(f_\theta(x), y)]$$

where f_θ and h_ϕ represent the predictor and data value estimator, respectively, with θ and ϕ as their corresponding parameters. The predictor and data value estimator are both implemented as multilayer perceptrons (MLPs) with a few fully connected layers to reduce computational costs. R_ϕ denotes the performance of the predictor on the validation dataset and serves as the reward function in reinforcement learning, while $\mathcal{L}$ is the training loss function.

The outline of the DVRL training procedure is as follows:

1. Compute the value for each training sample $p_n \in [0, 1]$ using the data value estimator, where p_n is given by $p_n = h_\phi(x_n, y_n)$, and x_n and y_n denote the input and corresponding label respectively of the n-th sample. A sigmoid activation is used on the output layer of h_ϕ.

2. Referring to the estimated value p_n as the probability of selecting the data, determine the selection indicator s_n by sampling from a Bernoulli distribution: $s_n \sim \text{Bernoulli}(p_n)$.
3. The predictor f_θ is trained using only the selected training samples.
4. Evaluate the performance of the trained predictor on the validation dataset and update the parameters ϕ of the data value estimator using the obtained reward R_ϕ. The parameter update is performed using the REINFORCE algorithm [56]: $\phi \leftarrow \phi + \alpha \nabla_\phi R_\phi$, where α is the learning rate, and the gradient of the reward function is given by:

$$\nabla_\phi R_\phi = R_\phi \cdot \nabla_\phi \log P\left(\{s\} \mid \phi\right). \tag{5}$$

   In this equation, $P\left(\{s\} \mid \phi\right)$ represents the likelihood of data selection indicators computed as:

$$\prod_{n=1}^{N} \left(h_\phi(x_n, y_n)\right)^{s_n} \left(1 - h_\phi(x_n, y_n)\right)^{1-s_n}.$$

5. Repeat the above steps iteratively to train the data value estimator h_ϕ.

See the original paper [15] for further technical details.

4. Proposed Method

This study proposes a method that applies the above three data valuation methods to detect and remove noisy training samples for training neural AES models. The proposed method consists of two main steps:

1. Estimating the value of each sample in the scored essay dataset based on a data valuation technique.

2. Removing potentially noisy training samples based on the estimated values and constructing an AES model.

The details of step (1) are described in Section 4.2, while those of step (2) are described in Section 4.3. Furthermore, an overview of the proposed AES training pipeline employing data valuation is illustrated in Fig. 1.

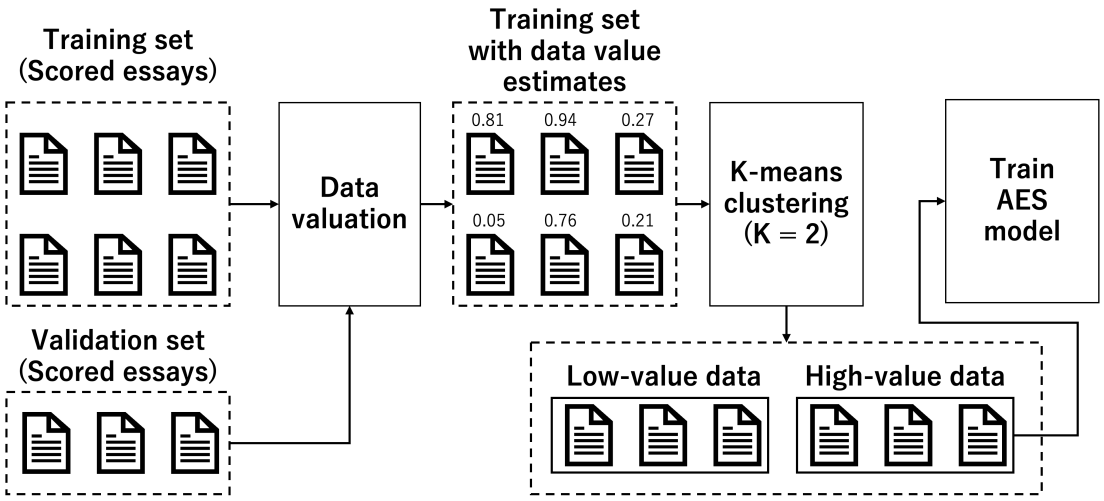

Figure 1: Overview of the proposed AES training pipeline using data valuation.

4.1. Task Setting

This study assumes that the given dataset is pre-split into a training dataset and a validation dataset. Specifically, let the given dataset be $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$, where x_n denotes the n-th essay for a specific essay prompt, y_n is its corresponding gold-standard score, and N represents the total number of samples. The score y_n takes an ordered categorical value $k \in \mathcal{K} = \{1, \ldots, K\}$, where K denotes the number of score categories for the given prompt. We assume that the dataset is divided into training $\mathcal{D}^t$ and validation $\mathcal{D}^v$ subsets. Note that, because the validation dataset $\mathcal{D}^v$ serves as the basis for identifying noisy samples in the training data, the validation dataset must be of high quality. Therefore, we assume that the accuracy of the scores assigned to essays in $\mathcal{D}^v$ has been verified in advance.

The data valuation process assigns data values to the training samples in $\mathcal{D}^t$ by evaluating the performance of an AES model on the validation dataset $\mathcal{D}^v$. We then aim to construct a highly performing AES model by using high-value training data. It is important to note that the architecture of the AES model used as a predictor during the data valuation process does not necessarily have to be the same as the final AES model.

4.2. Data Valuation for AES

As described above, we apply the three conventional data valuation methods to the AES task. To achieve this, we design a predictor AES model, which is required for all data valuation methods, and a data value estimator, which is specifically needed for DVRL. The following sections provide detailed descriptions of their architectures.

4.2.1. Architecture of Predictor AES

All the data valuation methods require constructing an AES model as a predictor and evaluating its performance on the validation dataset. Since the predictor is trained repeatedly during the data valuation process, it is essential to use a computationally efficient model.

Therefore, as the predictor, this study employs an MLP that consists of two fully connected layers, with the final output layer applying a sigmoid activation function, as defined in Eq. (1). The hidden layer dimension of the middle layer is set to 512. The model input consists of distributed representation vectors of essay texts obtained from the pre-trained decoding-enhanced BERT with disentangled attention (DeBERTa) model[1].

Model training during the data valuation process is conducted using Adam [57] with backpropagation and the MSE loss function. The learning rate is set to 1×10^{-3}, and the model is trained for 100 epochs. Furthermore, the AES performance of the trained model on the validation dataset is evaluated using quadratic weighted kappa (QWK), a widely adopted metric in AES research [5, 6, 58].

4.2.2. Data Value Estimator in DVRL

DVRL requires a data value estimator in addition to the predictor. Since the data value estimator is trained repeatedly alongside the predictor, it must also be computationally efficient.

Therefore, we construct the data value estimator as an MLP consisting of five fully connected layers, with a linear output layer using a sigmoid activation function. The hyperparameters are set to match those used in [15]. The input to the data value estimator consists of distributed representation vectors of essays obtained from the pre-trained DeBERTa model, along with their corresponding gold-standard scores.

4.3. Construction of the Final AES Model Using High-Value Data

To detect noisy data based on the calculated data values, we apply a clustering approach. Specifically, a K-means clustering algorithm with $K = 2$ is used to partition the dataset into a high-value cluster

[1]We used `microsoft/deberta-v3-large` from Hugging Face in our experiments.

Table 1
Summary of the ASAP dataset

Prompt	1	2	3	4	5	6	7	8
Num. of essays	1,783	1,800	1,726	1,772	1,805	1,800	1,569	723
Avg. length	350	350	150	150	150	150	250	650
Score range	2-12	1-6	0-3	0-3	0-4	0-4	0-30	0-60

and a low-value cluster. The final AES model is then trained using the essay samples in the high-value cluster.

As the final AES model constructed from the selected data, we adopt the BERT-based model[2] introduced in Section 2.3. The optimization is performed using AdamW, with the number of training epochs set to 10.

5. Evaluation Experiments

This section describes the experiments conducted to evaluate the effectiveness of the proposed method.

5.1. Experimental Procedure

In this experiment, we used the automated student assessment prize (ASAP) dataset, a benchmark dataset for AES. The ASAP dataset consists of scored English essays written in response to eight different essay prompts. Table 1 provides an overview of the dataset.

Using this dataset, we conducted the following experimental procedure:

1. The scored essay dataset for each prompt in the ASAP dataset was split into 60% training data, 20% validation data, and 20% test data.
2. For 20% of the training samples randomly selected, a perturbation was applied to the corresponding normalized score by adding or subtracting a value randomly sampled from the range [0.4, 0.6]. If the resulting score exceeded the range [0, 1], it was rounded to fit within the range.
3. The proposed method was applied to the training data containing noisy samples to select high-value, noise-free samples.
4. A BERT-based AES model was trained using the selected high-value samples, and its prediction accuracy was evaluated using the test dataset, with QWK as the evaluation metric.
5. While changing the data partitioning process in step 1, we repeated steps 1 to 4 five times. The final prediction accuracy was obtained as the average of these runs.

For comparison, we also conducted the same experiments without applying data selection in step 3, where the BERT-based AES model was trained using all training data with artificially generated noisy samples along with the validation data.

5.2. Experimental Results

The experimental results are shown in Table 2 with the highest accuracy for each setting highlighted in bold. The results indicate that the proposed methods consistently outperform the model trained on all data, regardless of the data valuation method applied. This demonstrates that removing noisy data by using the proposed approach effectively improves AES accuracy.

Furthermore, the average accuracy follows the order DVRL $>$ GS $>$ LOO, suggesting that DVRL provides the most accurate data valuation among the tested methods.

Table 2
Comparison of scoring accuracy (QWK) with and without the proposed method

	Prompt								Avg.
	1	2	3	4	5	6	7	8	
All data	0.529	0.403	0.501	0.692	0.655	0.673	0.521	0.308	0.535
LOO	0.621	0.410	0.549	0.705	0.688	0.675	0.662	0.410	0.590
GS	**0.787**	0.641	0.649	**0.749**	**0.792**	**0.735**	0.776	0.665	0.724
DVRL	0.781	**0.644**	**0.678**	0.738	0.786	**0.735**	**0.821**	**0.700**	**0.735**

Table 3
Detection accuracy of noisy data (F1-scores)

	Prompt								Avg.
	1	2	3	4	5	6	7	8	
LOO	0.527	0.381	0.413	0.362	0.354	0.352	0.635	0.449	0.434
GS	0.970	0.900	0.574	0.689	0.817	0.819	0.775	0.936	0.810
DVRL	**0.963**	**0.956**	**0.836**	**0.771**	**0.844**	**0.829**	**0.947**	**0.990**	**0.892**

Table 4
Computation time (units: seconds)

	Prompt								Avg.
	1	2	3	4	5	6	7	8	
LOO	**53**	**53**	**50**	**50**	**51**	**51**	**44**	**20**	**46**
GS	4,726	4,808	4,554	4,683	4,752	4,742	4,152	1,916	4,292
DVRL	106	106	106	106	107	107	106	100	105

5.3. Validation of Estimated Data Values

To evaluate the effectiveness of the proposed method in detecting noisy data, we assessed the accuracy of noisy data detection using the F1-score. Specifically, we treated the low-value cluster, obtained through clustering based on the estimated data values, as noisy data, and measured how well the proposed method identified these samples.

The results are shown in Table 3, showing that DVRL achieved the highest detection accuracy, followed by GS and LOO. This trend is consistent with the AES accuracy reported earlier, suggesting that higher data valuation accuracy contributed to improved AES performance.

5.4. Comparison of Computation Time

Table 4 shows the average time required for data valuation by each method, indicating that GS incurred substantially higher computational costs, while LOO and DVRL were almost comparable. This result also suggests that DVRL is a reasonable choice in terms of both computational efficiency and accuracy.

6. Application of the Proposed Method to Data Augmentation

As discussed in Section 1, the proposed method is expected to be effective when combined with data augmentation approaches. Data augmentation refers to techniques that artificially expand a training dataset [18] and is commonly employed when the available training samples are limited. With recent advancements in LLMs such as ChatGPT, LLM-based data augmentation has been explored in various NLP tasks [18, 19, 20, 21, 22, 23, 24]. However, the quality of augmented data is not always guaranteed

[2]We used `google-bert/bert-base-uncased` from Hugging Face as the basis BERT in our experiments.

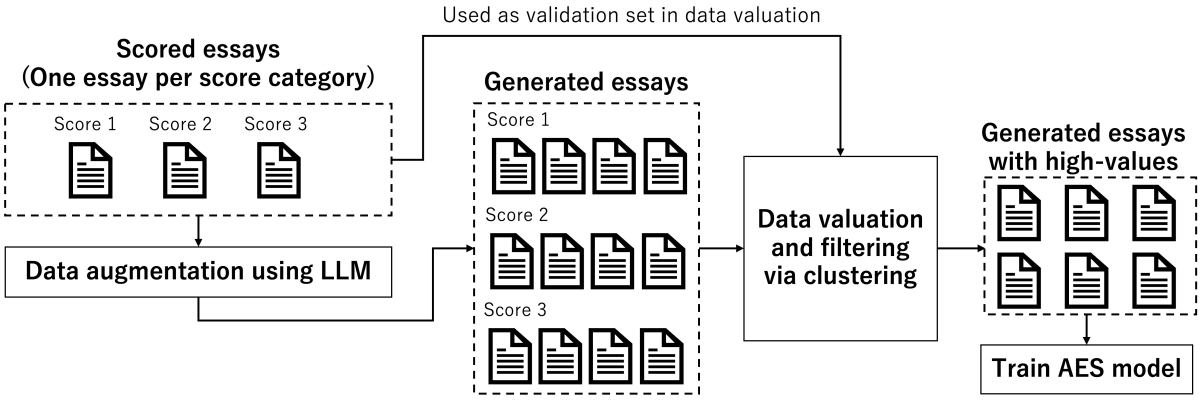

Figure 2: An overview of the proposed AES training method, which combines data augmentation using LLMs with data valuation-based filtering.

to be high, and there is a high likelihood of introducing noisy data. To address this issue, we propose applying our method to an augmented dataset and evaluating its effectiveness.

6.1. Task Settings and Methodology

In this section, we assume that the given dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$ contains a very limited number of samples. More specifically, we consider a scenario where only one essay per score category is available, meaning that $N = K$. Based on this dataset, we aim to generate scored essays using a data augmentation method. The overall process is outlined in Fig. 2.

The data augmentation is conducted using ChatGPT with a detailed instruction to generate an essay with a specified score by referencing the given pairs of essay texts and scores for all K score categories in $\mathcal{D}$ as few-shot samples. The model used for augmentation was `gpt-4o-mini-2024-07-18` [59], with the temperature parameter set to 1.1. The prompt given to ChatGPT is shown in Table 5.

Using the augmented data as the training dataset $\mathcal{D}^t$ and the original human-labeled data as the validation dataset $\mathcal{D}^v$ (namely, $\mathcal{D}^v = \mathcal{D}$), we apply our proposed method to identify and remove noisy augmented data. The final BERT-based AES model is then trained using the selected high-value samples from the training dataset.

6.2. Experiments Using Augmented Data

We evaluated the effectiveness of the above method by assessing scoring accuracy for each of the eight essay prompts in the ASAP dataset through five-fold cross-validation. As the data valuation method, we

Table 5
Prompt given to ChatGPT

Your task is to generate an essay with a score of "{target_score}" based on the characteristics identified from a given set of scored essays. The provided essays were written by American students in grades 7 to 10, each assigned a score ranging from {min_score} to {max_score}. Use these essays as a reference to generate new essays that align with the characteristics of the target score: {target_score}. Specifically, imitate features such as content, word count, grammatical and logical structure, wording, and spelling mistakes to match the characteristics of the target score category. ## Output each essay in the following format: - **score**: [The score category that the essay represents.] - **essay**: [The generated essay as a single string without any line breaks.] ## Examples: [Few-shot examples (pairs of essays and scores in the format described above) are given here]

Table 6
Evaluation results of data augmentation approach using the proposed method

Type of training data used	Prompt								Avg.
	1	2	3	4	5	6	7	8	
Original data $\mathcal{D}$	0.720	0.473	0.056	0.024	0.516	0.421	**0.732**	**0.676**	0.452
All augmented data	0.693	0.591	0.508	0.544	0.699	0.546	0.673	0.645	0.612
Filtered augmented data	**0.748**	**0.604**	**0.553**	**0.554**	**0.753**	**0.559**	0.646	0.649	**0.633**

employed the DVRL approach, which demonstrated the best performance in previous experiments. The number of augmented samples for each score category was set to 100. The accuracy metric was QWK.

For comparison, we conducted the same experiments without data selection, where all augmented data were used as the training data. In addition, we evaluated the accuracy of the BERT-based AES model trained only on the dataset $\mathcal{D} = \mathcal{D}^v$, which contains $N = K$ human-labeled essays, through the same prompt-wise five-fold cross-validation.

6.3. Experimental Results

The experimental results are presented in Table 6, which show that the use of augmented data positively impacted scoring accuracy on average. Furthermore, selecting augmented data through the proposed method further improved accuracy. This suggests that some augmented data may have inconsistencies between the assigned scores and actual essay quality, which can negatively impact the training process of the AES model. By effectively filtering out such unreliable augmented samples and retaining only high-quality ones, the proposed method is expected to enhance prediction accuracy.

The conventional approach using only original data $\mathcal{D}$ achieved higher accuracy on prompts 7 and 8. One possible explanation is that these prompts have a larger number of score categories, allowing the model to achieve sufficient performance with just a single essay per score category. Another possible explanation is that the wider score range may have made it challenging for ChatGPT to accurately capture subtle differences in essays corresponding to each score category during data augmentation. Consequently, the proportion of noisy data increased, making it difficult to retain high-value data even when the our method was applied.

In summary, the experiments demonstrate that applying the proposed method with a data augmentation approach is likely to be effective in cases where labeled data is substantially limited.

7. Conclusion

In this study, we proposed a method for detecting and removing noisy data from the training dataset by utilizing data valuation techniques to construct an accurate AES model. Experimental results demonstrated that the proposed method effectively filtered out noisy data, thereby enhancing the accuracy of AES models. Furthermore, we investigated the integration of our method with a data augmentation approach. The experimental results indicated that applying the proposed method allowed for the selective utilization of beneficial augmented data, leading to improved AES performance, in scenarios with very limited training data.

While the proposed method demonstrated effectiveness in filtering out noisy data and improving AES accuracy, the experimental conditions in this study were limited, and further investigations are needed to evaluate the generalizability of the proposed method. More specifically, in the simulation, strong noise was introduced into 20% of the training data, with score perturbations ranging from 40% to 60% of the score range. However, more detailed analysis is required to determine the extent to which the proposed method can detect noisy data under varying noise intensities and proportions. Additionally, in the data augmentation experiments, we assumed a setting where only one essay per score category was available. It is necessary to evaluate the performance of the proposed method under more diverse

conditions. Furthermore, this study explored a simple data augmentation approach using ChatGPT. To better understand the relative impact of different augmentation methods, future research should investigate the effectiveness of the proposed method using alternative data augmentation techniques.

In addition to the aspects discussed above, there remain several limitations that should be addressed in future work. First, the present study was conducted using only the English ASAP dataset under a holistic scoring setting. To evaluate the generalizability of the proposed method, it is necessary to examine its effectiveness in other scenarios, including analytic scoring, cross-prompt evaluation, and essays written in different languages. Second, although we compared three representative data valuation methods, we did not include other potentially relevant approaches, such as MentorNet [60] and Meta-Weight-Net [16], which apply data-weighting strategies during training. Comparison with these approaches would provide further insights into the relative advantages and limitations of our method. Third, the data values assigned by the proposed method are represented as scalar scores, making it difficult to interpret the reasons for data exclusion in linguistic or pedagogical terms. This lack of interpretability may raise concerns regarding fairness and transparency, particularly in high-stakes educational applications. Finally, the current method employs a fixed K-means clustering with $K = 2$ to separate high- and low-value data. Since the choice of thresholding strategy may affect model performance, it would be worthwhile to explore sensitivity analyses and alternative clustering approaches.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers 23K20727 and 24H00739.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4o in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Y. Abosalem, Assessment techniques and students' higher-order thinking skills, International Journal of Secondary Education 4 (2016) 1–11.

[2] E. R. Lai, Critical thinking: A literature review research report, Pearson's Research Reports 6 (2011) 40–41.

[3] O. L. Liu, L. Frankel, K. C. Roohr, Assessing critical thinking in higher education: Current state and directions for next-generation assessment, ETS Research Report Series 2014 (2014) 1–23.

[4] M. Uto, M. Okano, Learning automated essay scoring models using item-response-theory-based scores to decrease effects of rater biases, IEEE Transactions on Learning Technologies 14 (2021) 763–776.

[5] Z. Ke, V. Ng, Automated essay scoring: A survey of the state of the art, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 6300–6308.

[6] M. Uto, A review of deep-neural automated essay scoring models, Behaviormetrika 48 (2021) 459–484.

[7] M. Uto, Y. Xie, M. Ueno, Neural automated essay scoring incorporating handcrafted features, in: Proceedings of the 28th international conference on computational linguistics, 2020, pp. 6077–6088.

[8] F. Nadeem, H. Nguyen, Y. Liu, M. Ostendorf, Automated essay scoring with discourse-aware neural models, in: Proceedings of the 14th workshop on innovative use of NLP for building educational applications, 2019, pp. 484–493.

[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Annual Conference of the North American

Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186.
[10] Y. Wang, C. Wang, R. Li, H. Lin, On the use of BERT for automated essay scoring: Joint learning of multi-scale essay representation, arXiv (2022).
[11] M. Yamaura, I. Fukuda, M. Uto, Neural automated essay scoring considering logical structure, in: Proceedings of the 24th International Conference on Artificial Intelligence in Education, 2023, pp. 268–278.
[12] R. D. Cook, Detection of influential observation in linear regression, Technometrics 19 (1977) 15–18.
[13] S. C. S. H. S. Lim, K. Brain, ChoiceNet: Robust learning by revealing output correlations, arXiv (2018).
[14] A. Ghorbani, J. Zou, Data Shapley: Equitable valuation of data for machine learning, in: Proceedings of the 36th International conference on machine learning, 2019, pp. 2242–2251.
[15] J. Yoon, S. O. Arik, T. Pfister, Data valuation using reinforcement learning, in: Proceedings of the 37th International Conference on Machine Learning, volume 119, 2020, pp. 10842–10851.
[16] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, D. Meng, Meta-Weight-Net: Learning an explicit mapping for sample weighting, Advances in neural information processing systems 32 (2019).
[17] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, Journal of big data 6 (2019) 1–48.
[18] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, E. Hovy, A survey of data augmentation approaches for NLP, arXiv (2021).
[19] K. Cochran, C. Cohn, J. F. Rouet, P. Hastings, Improving automated evaluation of student text responses using GPT-3.5 for text data augmentation, in: Proceedings of 24th International Conference on Artificial Intelligence in Education, 2023, pp. 217–228.
[20] J. Wei, K. Zou, EDA: Easy Data Augmentation techniques for boosting performance on text classification tasks, arXiv (2019).
[21] V. Kumar, A. Choudhary, E. Cho, Data augmentation using pre-trained transformer models, arXiv (2020).
[22] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, Y. Qi, TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP, arXiv (2020).
[23] C. Shorten, T. M. Khoshgoftaar, B. Furht, Text data augmentation for deep learning, Journal of big Data 8 (2021) 1–34.
[24] H. Dai, Z. Liu, W. Liao, X. Huang, Y. Cao, Z. Wu, L. Zhao, S. Xu, W. Liu, N. Liu, et al., AugGPT: Leveraging ChatGPT for text data augmentation, arXiv (2023).
[25] S. Li, V. Ng, Automated essay scoring: A reflection on the state of the art, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 17876–17888.
[26] H. Misgna, B.-W. On, I. Lee, G. S. Choi, A survey on deep learning-based automated essay scoring and feedback generation, Artificial Intelligence Review 58 (2025) 1–40.
[27] R. Ridley, L. He, X. Dai, S. Huang, J. Chen, Prompt agnostic essay scorer: a domain generalization approach to cross-prompt automated essay scoring, arXiv (2020).
[28] F. S. Mim, N. Inoue, P. Reisert, H. Ouchi, K. Inui, Unsupervised learning of discourse-aware text representation for essay scoring, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 2019, pp. 378–385.
[29] M. A. Hussein, H. A. Hassan, M. Nassef, A trait-based deep learning automated essay scoring system with adaptive feedback, International Journal of Advanced Computer Science and Applications 11 (2020) 287–293.
[30] S. Mathias, P. Bhattacharyya, Can neural networks automatically score essay traits?, in: Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, 2020, pp. 85–91.
[31] T. Shibata, M. Uto, Analytic automated essay scoring based on deep neural networks integrating multidimensional item response theory, in: Proceedings of the 29th International Conference on Computational Linguistics, 2022, pp. 2917–2926.

[32] Y. Chen, X. Li, PMAES: Prompt-mapping contrastive learning for cross-prompt automated essay scoring, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2023, pp. 1489–1503.
[33] T. Shibata, M. Uto, Enhancing cross-prompt automated essay scoring by selecting training data based on reinforcement learning, in: Proceedings of the First Workshop on Automated Evaluation of Learning and Assessment Content, co-located with the 25th International Conference on Artificial Intelligence in Education, 2024.
[34] Y. Cao, H. Jin, X. Wan, Z. Yu, Domain-adaptive neural automated essay scoring, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, 2020, pp. 1011–1020.
[35] Z. Jiang, T. Gao, Y. Yin, M. Liu, H. Yu, Z. Cheng, Q. Gu, Improving domain generalization for prompt-aware essay scoring via disentangled representation learning, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2023, pp. 12456–12470.
[36] E. Amorim, M. Cançado, A. Veloso, Automated essay scoring in the presence of biased ratings, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2018, pp. 229–237.
[37] M. Dascalu, W. Westera, S. Ruseti, S. Trausan-Matu, H. Kurvers, ReaderBench learns Dutch: Building a comprehensive automated essay scoring system for Dutch language, in: Proceedings of the International Conference on Artificial Intelligence in Education, 2017, pp. 52–63.
[38] M. D. Shermis, J. C. Burstein, Automated Essay Scoring: A Cross-disciplinary Perspective, Routledge, 2002.
[39] D. Alikaniotis, H. Yannakoudakis, M. Rei, Automatic text scoring using neural networks, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2016, pp. 715–725.
[40] Y. Farag, H. Yannakoudakis, T. Briscoe, Neural automated essay scoring and coherence modeling for adversarially crafted input, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2018, pp. 263–271.
[41] C. Jin, B. He, K. Hui, L. Sun, TDNN: A two-stage deep neural network for prompt-independent automated essay scoring, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 1088–1097.
[42] M. Mesgar, M. Strube, A neural local coherence model for text quality assessment, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4328–4339.
[43] K. Taghipour, H. T. Ng, A neural approach to automated essay scoring, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1882–1891.
[44] Y. Wang, Z. Wei, Y. Zhou, X. Huang, Automatic essay scoring incorporating rating schema via reinforcement learning, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2018, pp. 791–797.
[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[46] E. Mayfield, A. W. Black, Should you fine-tune BERT for automated essay scoring?, in: Proceedings of Workshop on Innovative Use of NLP for Building Educational Applications, 2020, pp. 151–162.
[47] J. Xue, X. Tang, L. Zheng, A hierarchical BERT-based transfer learning approach for multi-dimensional essay scoring, IEEE Access 9 (2021) 125403–125415.
[48] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners, Advances in Neural Information Processing Systems 35 (2022) 22199–22213.
[49] S. Lee, Y. Cai, D. Meng, Z. Wang, Y. Wu, Unleashing large language models' proficiency in zero-shot essay scoring, in: Findings of the Association for Computational Linguistics, 2024, pp. 181–198.
[50] W. A. Mansour, S. Albatarni, S. Eltanbouly, T. Elsayed, Can large language models automatically score proficiency of written essays?, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, 2024, pp. 2777–2786.
[51] M. Stahl, L. Biermann, A. Nehring, H. Wachsmuth, Exploring LLM prompting strategies for joint

essay scoring and feedback generation, in: Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, 2024, pp. 283–298.
[52] Y. Wang, R. Hu, Z. Zhao, Beyond agreement: Diagnosing the rationale alignment of automated essay scoring methods based on linguistically-informed counterfactuals, in: Findings of the Association for Computational Linguistics, 2024, pp. 8906–8925.
[53] K. P. Yancey, G. Laflair, A. Verardi, J. Burstein, Rating short L2 essays on the CEFR scale with GPT-4, in: Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications, 2023, pp. 576–584.
[54] L. S. Shapley, A value for n-person games, Contribution to the Theory of Games 2 (1953).
[55] P. Dubey, On the uniqueness of the Shapley value, International Journal of Game Theory 4 (1975) 131–139.
[56] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning 8 (1992) 229–256.
[57] D. P. Kingma, Adam: A method for stochastic optimization, arXiv (2014).
[58] M. Uto, Y. Takahashi, Neural automated essay scoring for improved confidence estimation and score prediction through integrated classification and regression, in: Proceedings of the 25th International Conference on Artificial Intelligence in Education, 2024, pp. 444–451.
[59] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al., GPT-4o system card, arXiv (2024).
[60] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, L. Fei-Fei, Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels, in: International conference on machine learning, 2018, pp. 2304–2313.