Integration of Prediction Scores from Various Automated Essay Scoring Models Using Item Response Theory

Masaki Uto, Itsuki Aomi, Emiko Tsutsumi and Maomi Ueno

***Abstract*—In automated essay scoring (AES), essays are automatically graded without human raters. Many AES models based on various manually designed features or various architectures of deep neural networks have been proposed over the past few decades. Each AES model has unique advantages and characteristics. Therefore, rather than using a single AES model, appropriate integration of predictions from various AES models is expected to achieve higher scoring accuracy. In the present paper, we propose a method that uses item response theory to integrate prediction scores from various AES models while taking into account differences in the characteristics of scoring behavior among models. It is found that the proposed method achieves higher accuracy than that of individual AES models and conventional score-integration methods. Furthermore, the proposed method facilitates interpreting each AES model's scoring characteristics and score-integration mechanism.**

***Index Terms*—Automated essay scoring, item response theory, deep neural networks, rater effects**

I. Introduction

ESSAY-writing tests have been used in various assessment situations to measure examinees' practical and higher-order abilities, including logical thinking, critical reasoning, and creative thinking [1]–[5]. Essay-writing tests require grading by human raters for essays that are written by examinees concerning a given topic. However, essay grading is an expensive and time-consuming task, especially for large-scale tests [5], [6]. To resolve this problem, various studies have examined automated essay scoring (AES), in which natural language processing (NLP) and machine learning are used to grade essays automatically as an alternative to human grading. AES is also important in the context of teaching writing in an educational setting. To efficiently cultivate students' writing skills, immediate and accurate feedback on their writing is required [7], [8]. Particularly, the accuracy of the feedback is critical because erroneous feedback might lead to misconceptions and biases in students' knowledge and understanding [7]. Furthermore, many formative feedback systems and analytical scoring systems for students' writing, which are tools for supporting writing education, are strongly related to AES technologies (e.g., [8]–[12]). These facts indicate that realizing accurate AES is crucial for both teaching and assessing writing in an educational setting.

Two approaches are generally used in most AES models: *feature engineering* and *automatic feature extraction* [5], [6], [13]. The feature-engineering approach uses manually designed features, such as the essay length and the number of spelling errors, and predicts essay scores based on a regression or classification model with such feature values as input. A representative model is *e-rater* [14], [15], which was developed and used by the Educational Testing Service (ETS) organization. Many other models with various textual features have also been proposed in the past few decades (e.g., [16]–[26]).

Feature-engineering approach models are advantageous in terms of interpretability and explainability, but they generally require careful feature design and selection to achieve high accuracy. The automatic feature extraction approach has become popular to eliminate the need for feature engineering.

Recent automatic feature extraction approach models generally use deep neural networks (DNNs). An early DNN-AES model was proposed in 2016 by Taghipour and Ng [27]. Their model consisted of a convolutional neural network (CNN) and a recurrent neural network (RNN). Based on this model, various extension models intended to capture more complex textual features have been proposed [28]–[35]. For example, some extension models are designed to explicitly capture textual coherence, which is an important factor in essay quality [31]–[35]. Furthermore, there are other DNN-AES models based on transformer networks [36] instead of CNNs and RNNs. Such models [37]–[44] generally use pre-trained transformer-based language models, including Bidirectional Encoder Representations from Transformers (BERT) [45].

These DNN-AES models predict a score from the sequence of words in the essay, meaning that no manually designed features are required. However, some recent studies have proposed hybrid models that incorporate manually designed features into DNN-AES models [10], [25], [42], [44], [46] and have reported that such a hybrid approach is effective for improving scoring accuracy.

Such conventional AES models have different characteristics of scoring behavior because each model employs different features or different DNN architectures. Therefore, rather than using a single AES model, integrating predictions from various AES models appropriately is expected to achieve

Manuscript received TBD; revised TBD; accepted TBD. Date of publication TBD; date of current version TBD. This work was supported by JSPS KAKENHI Grant Numbers 19H05663, 19K21751 and 20K20817. This article is a revised and expanded version of a paper that was presented at the 22nd International Conference on Artificial Intelligence in Education, which took place in Utrecht, Netherlands from June 14 to 18, 2021 [DOI: 10.1007/978-3-030-78270-2_9]. *(Corresponding author: Masaki Uto.)*

The authors are with the Graduate School of Informatics and Engineering, The University of Electro-Communications, Chōfu, Tokyo 182-8585, Japan (e-mail: {uto, aomi, tsutsumi, ueno}@ai.lab.uec.ac.jp).

higher scoring accuracy. A simple score-integration strategy is to calculate the average scores or majority vote scores. However, such simple methods might be inaccurate because they ignore differences in characteristics of scoring behavior among AES models. Another score-integration strategy is *stacking*, a popular ensemble learning approach [47]. A stacking-based AES can be designed as a supervised regression model, such as linear regression, support vector regression, and regression tree, which receives multiple AES scores as input and outputs integrated scores However, the stacking method using such popular regression models has the following drawbacks.

1) Those regression models are necessarily not efficient in modeling the scoring behaviors of human raters and AES models because they are not specialized in the essay scoring domain. The inefficient modeling may prevent maximizing the scoring accuracy.
2) Those models do not provide a clear meaning for their parameters, making it difficult to analyze the scoring behaviors of human raters and AES models in detail. The lack of this interpretability hinders our understanding of the score-integration mechanism and the characteristics of individual AES models.

To resolve these problems, this paper proposes a method to integrate scores from various AES models using item response theory (IRT) models incorporating raters' characteristic parameters [48]–[57]. Those IRT models have long been studied in the educational measurement field to realize accurate scoring while taking into account differences in rater characteristics, such as severity and consistency. Those models have been applied to various performance assessments, including essay writing tests, and have demonstrated their effectiveness in realizing accurate scoring and detailed analysis of rater characteristics [56]–[60]. This study applies such IRT models by regarding AES models as human raters to obtain integrated essay scores. Our experiments using an AES benchmark dataset demonstrate the effectiveness of the proposed method.

Note that another AES method that uses IRT incorporating rater parameters was recently proposed [61], [62]. However, the objective of that study was to obtain accurate gold-standard scores for essays, which are then used for AES model training. Gold-standard scores for training data are generally created by sharing the essay grading task among many human raters, although scores from some raters may be inaccurate and unreliable [63]–[66]. Thus, that study proposed the use of IRT to remove the effects of such unreliable raters from training data, indicating that the objectives and method are completely different from those in our study.

It should also be noted that, although this study focuses on AES, the proposed method can also be used for automated short-answer grading (ASAG) and other text-scoring tasks, for which many models with different characteristics have been developed. For example, there are many ASAG models [67], [68]; some are similar to AES models, but others are different. Representative models similar to AES models include *c-rater* [69], which is a representative feature-engineering approach model, and CNN-RNN-based and BERT-based DNN models [70]–[73]. Major differences between AES and ASAG models are 1) the importance of coherence is often emphasized in AES but not necessarily in ASAG and 2) reference answers are often used for ASAG [74]–[76] but not for AES. Although some similar models and task-dependent models exist for different scoring tasks, as explained above, the proposed method is applicable to those tasks for which many different scoring models exist.

II. Research Challenges

This study provides a theoretical contribution beyond the simple engineering application of the improved essay scoring system. The purpose of this study is to clarify the effectiveness of IRT models with rater characteristic parameters in order to integrate predictions from various AES models. To this end, we present the following two research challenges.

1) The proposed IRT-based score-integration method can integrate scores from various AES models while considering the characteristics of their scoring behavior, which are parameterized appropriately based on extensive research in the educational measurement domain. Owing to the sophisticated modeling of scoring behavior, the proposed method is expected to provide higher scoring accuracy compared with individual AES models and other score-integration methods, including the general stacking method. Accordingly, our first research challenge is to examine how effectively the proposed method improves scoring accuracy.
2) IRT models provide explicit meaning for the model parameters, helping us to understand the score-integration mechanism and the characteristics of individual AES models. Thus, our second research challenge is to show how to analyze the score integration mechanism and the characteristics of AES models based on the proposed method.

Although IRT models that incorporate rater-characteristic parameters have been widely used in various educational assessment studies (e.g., [49]–[57]), no previous study has used such IRT models to integrate predictions from various AES models. Therefore, it remains unclear how those IRT models might be applied to realize AES integration and how the method would be beneficial. The fact that our study answers these questions confirms its theoretical contribution.

III. Automated Essay Scoring Models

This section presents a brief review of conventional AES models based on the feature-engineering and automatic feature extraction approaches.

A. Feature-Engineering Approach

In the feature-engineering approach, models predict essay scores based on textual features, which human experts must design manually. Typical features are essay length and number of grammatical and spelling errors. In this approach, such textual features are first calculated from a target essay text, then the feature vector is input into a regression or classification model, and a score is output.

A representative model is e-rater [14], which was developed by ETS and has been used in the Test of English as a Foreign Language and the Graduate Record Examination. E-rater v.2 [15] uses 12 features and predicts essay scores based on a linear regression model with empirically determined weight parameters. The Enhanced AI Scoring Engine (EASE) [16][1] is another model that has recently come into widespread use and achieved high performance in the Automated Student Assessment Prize (ASAP) competition on Kaggle[2]. EASE uses several types of features, including length-based features, part-of-speech-based features, prompt-relevant features, and bag-of-words-based features. There are many other models that incorporate various types of features, such as word topicality [17], bag-of-super-word embedding [18], argument features (e.g., number of claims and number of supporting relations) created by argument-mining techniques [19], a sentence semantic similarity defined using a graph-based text analysis method [20], and semantic features that are specific to the Chinese language [21].

Feature-engineering approach models are generally based on linear regression [14], [16], support vector regression (SVR) [16], XGBoost [25], and DNNs [26] and require their training using a training dataset, although e-rater v.2 uses empirically determined weights for the regression model.

B. Automatic Feature Extraction Approach

As explained in Section I, recent automatic feature extraction approach models generally use DNNs. Many DNN-AES models have recently been proposed (e.g., [27]–[35], [37]–[44]).

One of the most popular DNN-AES models is the CNN-RNN-based model [27]. This model calculates a score for a targeted essay, which is defined as a sequence of words, through five DNN layers, namely, the lookup table layer, the convolution layer, the recurrent layer, the pooling layer, and the linear layer with sigmoid activation. See Appendix A for details on the whole architecture. There are many variants of this model, such as those employing different pooling methods in the pooling layer [28], [30], those using a different word embedding in the lookup table layer [28], and those consisting of a word-level CNN and a sentence-level CNN [29].

One limitation of the CNN-RNN-based models is that they cannot directly consider textual coherence, which represents the semantic connection and consistency of the whole text. Coherence is an important factor for determining the quality of essays. Thus, some DNN-AES models with a function to capture textual coherence explicitly have been proposed [31]–[35], [44]. A representative model is the SkipFlow model [32], an extension of the CNN-RNN-based model that incorporates a neural tensor layer, which explicitly captures textual coherence. See Appendix B for details on the SkipFlow model. Another model tries to capture the coherence based on the continuity in the semantics between the adjacent two sentences [33].

[1]https://github.com/edx/ease

[2]https://www.kaggle.com/c/asap-aes

While the above-introduced models used CNNs and RNNs, some recent models have used attention-based DNN architectures [37]–[42]. A popular attention-based DNN is a transformer network [36] that consists of stacked self-attention and fully connected layers. Transformer networks are known to capture long-distance dependency between words in a text with more accuracy than that of RNNs and CNNs.

Transformer-based DNN-AES models typically use pre-trained models. A representative pre-trained model is BERT [45], which was released by the Google AI Language team. BERT is pre-trained on massive amounts of unlabeled text data for two tasks, called *masked language modeling* and *next-sentence prediction*. BERT can be used for various NLP tasks, including AES, by applying a fine-tuning (model re-training) based on a task-specific supervised dataset. See Appendix C for details on the BERT-based AES. The BERT-based AES also has various extensions, such as those incorporating architectures to capture the textual coherence [43], [44], those extended toward multi-task learning [39], [77], and those using the DistilBERT [78], a variant of BERT [79].

C. Hybrid Approach

The feature-engineering and DNN-based automatic feature extraction approaches can be viewed as complementary rather than competing [6] because they have different advantages and drawbacks. Thus, some hybrid models that integrate the two approaches have recently been proposed [10], [25], [42], [44], [46].

Hybrid models are generally formulated as DNN-AES models incorporating manually designed features. Specifically, they concatenate a feature vector to either a predicted score of a DNN-AES model or a hidden vector obtained from an intermediate layer of a model. Then, the concatenated vector is mapped to a score value through a regression layer, such as a linear layer with sigmoid activation. The DNN-AES models used in the hybrid models include variants of the CNN-RNN-based model [10], [25] and the BERT-based model [42]. As an example, Appendix C introduces details on the BERT-based hybrid AES model.

Other hybrid models [44], [46] consist of two types of DNNs: one processes word sequences in the same way as the conventional DNN-AES model, and the other processes manually designed features.

IV. Item Response Theory

The conventional AES models discussed above have different scoring behaviors because they employ different features or different DNN architectures. The purpose of this study was to integrate prediction scores from various AES models using IRT while considering differences in the characteristics of their scoring behaviors.

IRT [80] is a test theory based on mathematical models. IRT uses probabilistic models, called IRT models, to estimate examinees' abilities from testing data, which generally consist of binary or polytomous scores that the examinees received on test items. IRT offers the following benefits: 1) Examinee

ability can be estimated in the context of test item characteristics, including item difficulty and discrimination. 2) Abilities of examinees who take different tests can be estimated on the same scale. 3) Missing data can be applied easily.

Traditional IRT models are applicable to data consisting of scores that examinees receive on test items. Examples include the Rasch model, the two-parameter logistic model, the graded response model [81], and the generalized partial credit model [82]. However, this study applied IRT to other data consisting of scores for each examinee's essay provided by multiple raters, including human raters and AES models. IRT models incorporating rater characteristic parameters can be applied to such data [49]–[57].

The most popular model is the many-facet Rasch model (MFRM) [51]. The MFRM, however, relies on some strong assumptions that do not hold in practice. Various extensions of the models have been proposed to relax these assumptions, including hierarchical rater models [52], [53], rater bundle models [54], and trifactor models [55]. The present study employs one of the latest extension models, which is called generalized MFRM (GMFRM) [57].

A. Generalized Many-Facet Rasch Model

In the GMFRM, the probability that rater r assigns score k to the essay of examinee j for test item i (which means an essay task or a prompt) is defined as

$$P_{ijrk} = \frac{\exp \sum_{m=1}^{k} \left[D\alpha_i \alpha_r (\theta_j - \beta_i - \beta_r - d_{rm}) \right]}{\sum_{l=1}^{K} \exp \sum_{m=1}^{l} \left[D\alpha_i \alpha_r (\theta_j - \beta_i - \beta_r - d_{rm}) \right]}, \tag{1}$$

where θ_j represents the latent ability of examinee j, α_i and β_i represent the respective discrimination power and difficulty of item i, α_r and β_r represent the respective consistency and severity of rater r, d_{rm} represents the severity of rater r against rating category m, and K indicates the number of categories. $D = 1.7$ is the scaling constant used to minimize the difference between the normal and logistic distribution functions. Here, $\sum_{i=1}^{I} \log \alpha_i = 0, \sum_{i=1}^{I} \beta_i = 0, d_{r1} = 0$, and $\sum_{k=2}^{K} d_{rk} = 0$ are given for model identification.

Note that this study applies the GMFRM to each item independently by removing the item parameters for the following two reasons.

1) To appropriately estimate the item parameters based on the original GMFRM while ensuring parameter linking, we require a scored essay dataset in which some examinees answered all the items [83], [84]. However, almost none of the existing datasets that are used for AES studies include such examinees. Therefore, it would be very difficult to estimate the item parameter appropriately based on the existing AES datasets.
2) Our objective is to estimate the integrated essay scores by using the GMFRM while considering the scoring behavior of each individual AES model, which is represented by the rater parameters in the model. Thus, the main interest in our use of IRT is the rater parameters, not the item parameters.

Although the item parameters are typically a major interest when using IRT, omitting them in this study is reasonable and does not impair the main feature of the IRT models that incorporate rater parameters, for the above reasons. When the item parameters are omitted, the GMFRM equation can be rewritten as follows:

$$P_{jrk} = \frac{\exp \sum_{m=1}^{k} \left[D\alpha_r (\theta_j - \beta_r - d_{rm}) \right]}{\sum_{l=1}^{K} \exp \sum_{m=1}^{l} \left[D\alpha_r (\theta_j - \beta_r - d_{rm}) \right]}. \tag{2}$$

In this form of GMFRM, θ_j represents not only the ability of examinee j, but also a latent score of the examinee's essay estimated from multiple raters' scores, because there is only a single essay for each examinee.

B. Interpretation of Rater Parameters in GMFRM

The GMFRM can consider the following three common rater characteristics [63], [85]–[89]:

- **Consistency:** The degree to which a rater assigns similar ratings to essays of similar quality.
- **Severity:** The tendency of a rater to give consistently lower ratings than are justified by the quality of the essays.
- **Range restriction:** The tendency to overuse a few rating categories.

To show how these characteristics are represented, Fig. 1 shows item response curves (IRCs) of the GMFRM, which are drawn by plotting the probability P_{jrk} in (2), for four raters for the parameters presented in Table I. In the figure, the horizontal axis shows the latent score θ_j and the vertical axis shows the probability P_{jrk}. These IRCs show that essays with higher θ_j tend to obtain higher scores.

In the GMFRM, rater consistency is represented by α_r, with lower values indicating smaller differences in response probabilities between rating categories. This can be confirmed in Fig. 1, which compares raters 1 and 2, who have different consistency levels. This figure suggests that scores given by a rater with a lower consistency parameter will be unreliable because the rater tends to assign different ratings to essays with similar qualities.

Rater severity is represented by β_r. The IRC shifts to the right as this parameter value increases, indicating that raters with high β_r values have a tendency to consistently assign low scores. In Fig. 1, the IRC for rater 3 with a high β_r value shifts to the right overall.

The GMFRM represents the range restriction characteristic as d_{rm}. The closer $d_{r(m+1)}$ and d_{rm} become, the lower the overall probability of responding with category m. Conversely, the higher the difference $d_{r(m+1)} - d_{rm}$ becomes, the higher the response probability for category m. In Fig. 1, rater 4 has smaller $d_{r3} - d_{r2}$ and $d_{r5} - d_{r4}$ values and relatively larger $d_{r4} - d_{r3}$ and $d_{r6} - d_{r5}$ values. Thus, in the IRC, response probabilities for categories 2 and 4 decrease, whereas those for categories 3 and 5 increase, representing a range restriction characteristic with overuse of categories 3 and 5 while avoiding categories 2 and 4.

The GMFRM can estimate latent scores θ_j while taking into account differences in these characteristics among raters, while earlier popular IRT models with rater parameters, including MFRM, cannot consider all the above rater characteristics

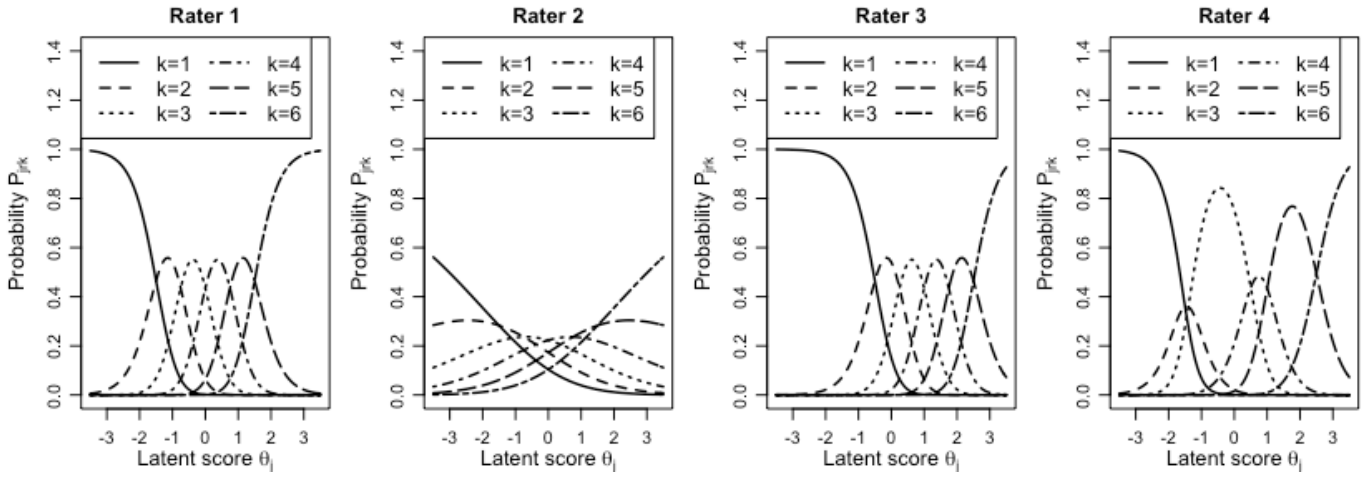

Fig. 1. Item response curves of four raters for the parameters presented in Table I.

TABLE I
PARAMETERS FOR FOUR RATERS WITH DIFFERENT CHARACTERISTICS.

Rater	α_r	β_r	d_{r2}	d_{r3}	d_{r4}	d_{r5}	d_{r6}
1	1.5	0.0	-1.50	-0.75	0.00	0.75	1.50
2	0.2	0.0	-1.50	-0.75	0.00	0.75	1.50
3	1.5	1.0	-1.50	-0.75	0.00	0.75	1.50
4	1.5	0.0	-1.50	-1.40	0.50	1.00	2.50

simultaneously. Thus, this model can realize an accurate score estimation compared with the other IRT models when raters with various characteristics exist [57], [58]. This is why we chose the GMFRM.

V. PROPOSED METHOD

This study proposes a method to integrate scores from various AES models using the GMFRM. Specifically, the proposed method applies the GMFRM by regarding AES models as human raters. The proposed method consists of three steps, namely, *AES model training*, *IRT parameter estimation*, and *integrated score prediction*. The detailed procedure for integrating these steps is as follows:

- **AES model training:** First, we train multiple AES models individually using training data consisting of essays with gold-standard human scores. This is the same as the procedure required to train any conventional AES model.
- **IRT parameter estimation:** This step estimates the characteristic parameters of the AES models based on the GMFRM. The parameter estimation is conducted using another dataset consisting of essays with gold-standard human scores, such as development data. The detailed procedure for this is as follows: 1) Generate prediction scores for essays in the data using each trained AES model. 2) Estimate the GMFRM parameters using those AES scores and the gold-standard human scores. Fig. 2 illustrates the outline of this procedure. Through this procedure, we can obtain the GMFRM parameters for the AES models and the human rater who created gold-standard scores. This study uses a Bayesian estimation based on a Markov-chain Monte Carlo (MCMC) algorithm for the IRT parameter estimation, as we detail in Section VI-B.
- **Integrated score prediction:** Using the trained AES models and their GMFRM parameters, this step predicts integrated scores for new essays. The outline of this procedure is illustrated in Fig. 3. As shown in the figure, we first generate prediction scores for the essays from the trained AES models individually. Then, the predicted scores are used to estimate the latent score θ_j for each essay based on the GMFRM. In this estimation, characteristic parameters of the AES models are given. Finally, the estimated latent scores θ_j are projected to an original rating scale on human rater criteria. Specifically, letting $r = 0$ be the human rater, the rescaled score y_j, which

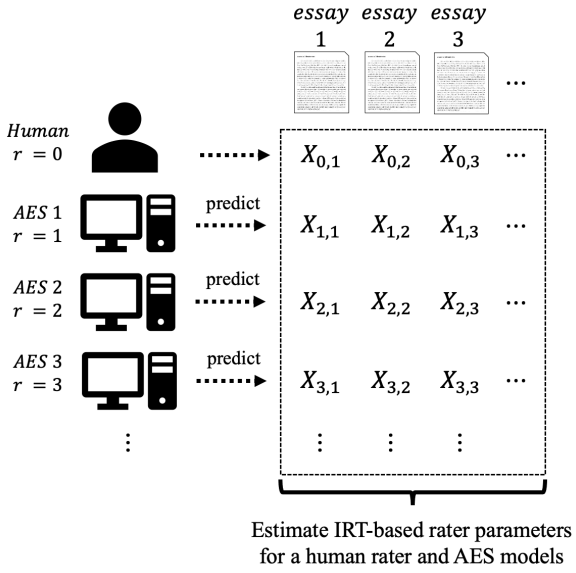

Fig. 2. Outline of IRT parameter estimation in the proposed method. Note that $X_{r,j}$ represents the score for the essay of the j-th examinee provided by the r-th AES model or human (where $r = 0$ represents the human rater and $r \geq 1$ corresponds to AES models).

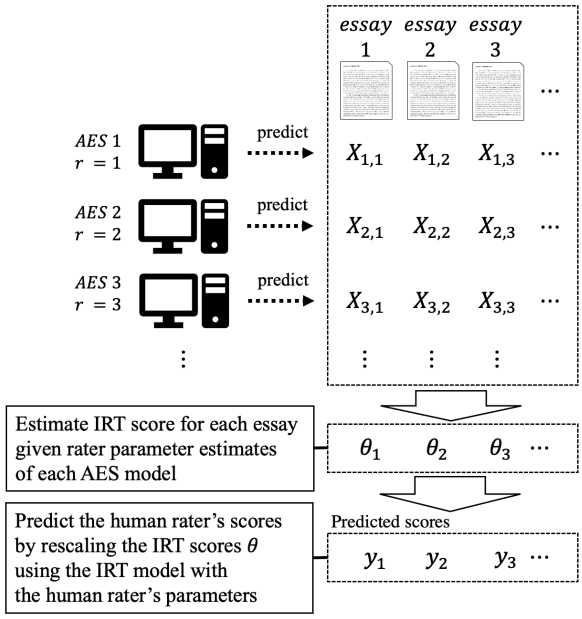

Fig. 3. Outline of integrated score prediction in the proposed method.

corresponds to the expected value of the human rater's score, is calculated as follows:

$$y_j = \sum_{k=1}^{K} k \cdot P_{j0k}. \tag{3}$$

Note that P_{j0k} is calculable based on Eq. (2) given the estimated latent scores θ_j and the human rater's parameters calibrated in the *IRT parameter estimation* step. This score rescaling is required for the following two reasons: 1) The latent scores θ_j are estimated on the logit scale [-∞, ∞], which differs from the original categorical score ranges. 2) The main goal of AES is to predict scores on the rating scale of the gold-standard human rater.

Algorithm 1 shows the detailed process of the proposed method. Here, we assume that training data and development data, which are composed of essays and gold-standard human scores, are available in the training phase. Furthermore, we assume the execution of AES for essays within test data. In Algorithm 1, $\boldsymbol{E}^{train}$, $\boldsymbol{E}^{dev}$, and $\boldsymbol{E}^{test}$ represent essays in the training data, development data, and test data, respectively. Furthermore, $\boldsymbol{X}^{train}$ and $\boldsymbol{X}^{dev}$ represent the gold-standard human scores in training data and development data, respectively, and R indicates the number of candidate AES models. The process in each function is as follows:

- $TrainAES\left(r, \boldsymbol{E}^{train}, \boldsymbol{X}^{train}, \boldsymbol{E}^{dev}, \boldsymbol{X}^{dev}\right)$ trains the r-th AES model using training data ($\boldsymbol{E}^{train}$, $\boldsymbol{X}^{train}$) and returns trained model M_r. Some models may use development data ($\boldsymbol{E}_{dev}$, $\boldsymbol{X}_{dev}$) for early stopping or hyperparameter tuning in the training phase.
- $PredAES(M_r, \boldsymbol{E})$ predicts scores for given essays $\boldsymbol{E}$ using trained model M_r and returns the prediction scores $\boldsymbol{X}_r$.

Algorithm 1 Algorithm of the proposed method

Require: $\boldsymbol{E}^{train}, \boldsymbol{E}^{dev}, \boldsymbol{E}^{test}, \boldsymbol{X}^{train}, \boldsymbol{X}^{dev}$

1: **for** $r \leftarrow 1$ to R **do**
2: $\quad M_r \leftarrow TrainAES\left(r, \boldsymbol{E}^{train}, \boldsymbol{X}^{train}, \boldsymbol{E}^{dev}, \boldsymbol{X}^{dev}\right)$
3: $\quad \boldsymbol{X}_r^{dev} \leftarrow PredAES(M_r, \boldsymbol{E}^{dev})$
4: **end for**
5: $\boldsymbol{\xi} \leftarrow EstIrtParam(\{\boldsymbol{X}^{dev}, \boldsymbol{X}_1^{dev}, \ldots, \boldsymbol{X}_R^{dev}\})$
6: **for** $r \leftarrow 1$ to R **do**
7: $\quad \boldsymbol{X}_r^{test} \leftarrow PredAES(M_r, \boldsymbol{E}^{test})$
8: **end for**
9: **for** $j \leftarrow 1$ to $|\boldsymbol{E}^{test}|$ **do**
10: $\quad \theta_j \leftarrow EstIrtScore(j, \boldsymbol{\xi}, \{\boldsymbol{X}_1^{test}, \ldots, \boldsymbol{X}_R^{test}\})$
11: $\quad y_j \leftarrow Eq.(3)$ given θ_j and human rater parameters in $\boldsymbol{\xi}$.
12: **end for**

- $EstIrtParam(\boldsymbol{X})$ runs the GMFRM parameter estimation using given score data $\boldsymbol{X}$ and returns estimated rater parameters $\boldsymbol{\xi}$, consisting of α_r, β_r, and d_{rm} for each AES model and human rater.
- $EstIrtScore(j, \boldsymbol{\xi}, \boldsymbol{X})$ computes the latent score θ_j from data $\boldsymbol{X}$ based on the GMFRM with the rater parameter estimates $\boldsymbol{\xi}$.

In Algorithm 1, line 2 corresponds to the AES model training procedure explained above. Lines 3 and 5 correspond to the above-explained IRT parameter estimation procedure. Also, line 6 and subsequent lines correspond to the integrated score prediction procedure.

Through the above procedures, the proposed method can output scores that integrate prediction scores from multiple AES models while considering the characteristics of their scoring behavior, and the output scores are projected onto the rating scale of the human rater.

Note that we can design a similar method based on factor analysis (FA) because FA and IRT are closely related [90], [91]. For example, some IRT models are known to be equivalent to some confirmatory categorical FA models with one factor [90]. Major differences between them are the purpose and domain [91]. The primary purposes of the IRT, which specializes in the educational and psychometric measurement domain, are scoring and test analysis, whereas the primary purpose of the FA, which is used in various contexts, is to investigate the construction of latent factors behind observed multivariate data. We used the IRT because its purpose and domain fit our study well, making the interpretation of the model parameters easy and natural.

Principal component analysis (PCA), which intends to estimate a latent factor behind observed data, would also be regarded as a similar approach to the IRT and FA. However, PCA cannot realize the score integration that we realized in the proposed method. Here, suppose we construct a PCA model using a dataset consisting of scores from multiple AES models and the gold-standard human rater, as in the IRT parameter estimation step. In that case, the constructed PCA model cannot calculate integrated scores for new essays because we have no gold-standard human scores for such essays. Furthermore, when we construct a PCA model using

only scores from multiple AES models, the scale of the model scores might not be consistent with the rating scale of the gold-standard human rater. Thus, we removed the PCA from the candidate pool for the proposed method.

VI. Experiments

In this section, we present evaluation results for the effectiveness of the proposed method based on experiments with actual data.

A. Dataset

Our experiments used the ASAP dataset, which has been published for Kaggle competitions. This dataset, which has commonly been used in various AES studies as benchmark data, consists of essays for eight prompts, written by students from grades 7 to 10. Each essay has one gold-standard score from a human rater. Scores are provided based on ordered categories with different value ranges. Each prompt corresponds to one of the three essay types: argumentative, source-dependent, and narrative [92]. The argumentative type asks students to discuss and justify their opinion on a specific topic, whereas the source-dependent type asks students to respond to a question about a given text. The narrative type asks students to narrate a story about a specific topic. See Table II for detailed statistics and types for each prompt.

The AES models are generally trained and evaluated for each prompt individually in many AES studies, so our experiments also follow this procedure.

B. Setup

Using the ASAP dataset, we evaluated scoring accuracy in each prompt based on five-fold cross-validation. In each partition, 60% of the data were used as the training data, 20% as the development data, and 20% as the test data. As the evaluation metric, we used the quadratic-weighted Kappa (QWK), which is a metric showing agreement between predicted scores and ground truth. The QWK is the common evaluation metric in the ASAP competition.

Our experiments used six AES models:

- **EASE (SVR), EASE (BLRR):** As a recently introduced popular feature-engineering approach model, we used the EASE model described in Subsection III-A. EASE typically uses Bayesian linear ridge regression (BLRR) and SVR as the regression models. We thus examined both variants of EASE. We implemented the models using scikit-learn [93] following the method in [16].
- **XGBoost:** As another feature-engineering approach model, we used a XGBoost model with the manually designed features proposed in [25]. A unique characteristic of this model is the use of parse-tree-based features, which are not used in EASE. We used CoreNLP [94] to generate parse trees and implemented the XGBoost model following the method in [95].
- **RNN:** As the most traditional DNN-based automatic feature extraction approach model, we used the CNN-RNN-based model detailed in Appendix A. Note that we omitted the optional convolution layer in our implementation. We implemented this model using PyTorch[3].
- **SkipFlow:** We also used the SkipFlow model detailed in Appendix B as another DNN-AES model with different characteristics. This is the most popular model incorporating a function that directly captures textual coherence, as explained in Section III-B. We used PyTorch to implement this model.
- **Hybrid–BERT:** We used the fine-tuned BERT model incorporating manually designed features [42], detailed in Appendix C, as a hybrid model. We used the uncased pre-trained *BERT-base* model and PyTorch for implementation.

We tokenized the essays using the NLTK tokenizer[4]. Other details, including hyperparameter settings, were the same as those used in the original studies.

We compared the proposed method with three common score-integration methods:

- **MEAN:** Arithmetic averaging of multiple AES scores.
- **VOTING:** Hard voting for multiple AES scores.
- **STACKING:** We examined four stacking models using a linear regression model, a Ridge regression model, an SVR, and a boosting model. We designed these models to receive multiple AES scores as input and predict a gold-standard human score. We used the scikit-learn library to implement these models. We trained these models using the development data in the same way as in the IRT parameter estimation of the proposed method.

Note that these three integration methods encompass most of the popular ensemble methods that integrate outputs from multiple models. This can be confirmed from the fact that conventional ensemble methods are commonly categorized as *weighting-based methods* or *meta-learning methods*, where the most popular weighting-based methods are *majority voting* and *output averaging* and the most popular meta-learning method is *stacking* [47].

We also examined some variants of the proposed method by changing the employed IRT models. As explained in Section IV-B, the GMFRM can represent the three common rater characteristics, namely, consistency, severity, and range restriction. Some GMFRM variants in which some rater parameters are restricted can be regarded as models equivalent to some earlier IRT models with rater parameters, including MFRM. For this

TABLE II
Statistics of the ASAP Dataset.

Prompt	# of Essays	Avg. length	Score range	Essay type
1	1,783	350	2-12	Argumentative
2	1,800	350	1-6	Argumentative
3	1,726	150	0-3	Source-Dependent
4	1,772	150	0-3	Source-Dependent
5	1,805	150	0-4	Source-Dependent
6	1,800	150	0-4	Source-Dependent
7	1,569	250	0-30	Narrative
8	723	650	0-60	Narrative

[3]https://pytorch.org/
[4]http://www.nltk.org/

TABLE III
QWK FOR EACH PROMPT. THE AVG. COLUMN SHOWS THE AVERAGE QWK VALUE FOR EACH METHOD. THE p-VALUE COLUMN SHOWS THE RESULTS OF THE ONE-TAILED PAIRED t-TEST BETWEEN THE PROPOSED METHOD USING GMFRM AND THE OTHER RESPECTIVE METHOD.

		Prompt								Avg.	p-value
		1	2	3	4	5	6	7	8		
Individual models	EASE (BLRR)	0.8038	0.6029	0.6555	0.7171	0.7845	0.7612	0.7300	0.6656	0.7151	<0.01
	EASE (SVR)	0.5578	0.5328	0.5644	0.5711	0.7397	0.6902	0.5451	0.3757	0.5721	<0.01
	XGBoost	0.8138	0.6397	0.5929	0.6596	0.7627	0.6573	0.6921	0.6704	0.6861	<0.01
	RNN	0.7769	0.6185	0.6511	0.7299	0.7542	0.7661	0.7496	0.5074	0.6942	<0.05
	SkipFlow	0.7984	0.6516	0.6568	0.7294	0.7841	0.7820	0.7512	0.6138	0.7209	<0.05
	Hybrid–BERT	0.8271	0.6372	0.6716	0.6204	0.7803	0.6728	0.7202	0.6723	0.7003	<0.01
Conventional integration methods	MEAN	0.8210	0.6771	0.6644	0.7185	0.7959	0.7725	0.7674	0.6722	0.7361	<0.01
	VOTE	0.8343	0.6620	0.6749	0.7287	0.7937	0.7710	0.7484	0.6700	0.7354	<0.05
	STACKING (Linear)	0.8313	0.6644	0.6492	**0.7386**	0.7861	0.7839	0.7701	0.6922	0.7395	<0.01
	STACKING (Ridge)	0.8316	0.6630	0.6477	**0.7386**	0.7867	0.7835	0.7703	0.6925	0.7392	<0.01
	STACKING (SVR)	0.8221	0.6230	0.6561	0.7235	0.7804	0.7704	0.7714	0.5810	0.7160	<0.05
	STACKING (Boosting)	0.8270	0.6599	0.6366	0.7367	0.7878	0.7838	0.7568	0.6439	0.7291	<0.05
Proposed method	GMFRM	**0.8365**	**0.6785**	0.6695	0.7375	**0.7972**	**0.7850**	0.7893	0.7095	**0.7562**	-
	Consistency-fixed GMFRM	0.8351	0.6657	**0.6755**	0.7223	0.7851	0.7608	**0.7979**	0.6902	0.7416	<0.05
	Severity-fixed GMFRM	0.8313	0.6673	0.6645	0.7380	0.7968	0.7734	0.7875	**0.7099**	0.7461	<0.05
	Threshold-fixed GMFRM	0.8309	0.6690	0.6505	0.7117	0.7905	0.7598	0.7716	0.6944	0.7348	<0.01
	MFRM	0.7944	0.6089	0.6630	0.6868	0.7769	0.7284	0.7710	0.6669	0.7120	<0.01

reason, we examined some restricted versions of the GMFRM, including MFRM, as detailed below.

- **Consistency-fixed GMFRM**: A GMFRM in which α_r is restricted to 1 for all raters $r \in \mathcal{R}$, meaning all raters share the same consistency level. This model is equivalent to the variant of MFRM shown in [48], [86].
- **Severity-fixed GMFRM**: A GMFRM in which β_r is restricted to 0 for all raters $r \in \mathcal{R}$, meaning all raters share the same severity level.
- **Threshold-fixed GMFRM**: A GMFRM in which d_{rm} is changed to d_m for all raters $r \in \mathcal{R}$, meaning no difference in range restriction characteristics exists among raters.
- **MFRM:** The most popular IRT model that incorporates rater parameters. MFRM is equivalent to a GMFRM in which α_r is restricted to 1 and d_{rm} is changed to d_m for all raters.

Although the severity-fixed GMFRM and the threshold-fixed GMFRM have no corresponding earlier IRT models, we examined them to investigate the effects of each rater parameter.

To estimate IRT parameters in the $EstIrtParam()$ function shown in Algorithm 1, we applied an expected a posteriori (EAP) estimation, a type of Bayesian estimation that is known to provide accurate estimations for complex IRT models [56], [96], using a MCMC algorithm. As the MCMC algorithm, we used the No-U-Turn sampler [97] based on the Hamiltonian Monte Carlo approach [98]. The estimation program was implemented in RStan [99], [100]. Following the original GMFRM paper [57], we calculated the EAP estimates using parameter samples obtained from 2000–4000 periods within three independent MCMC chains. Furthermore, the prior distributions were also the same as those used in [57], namely,

$$\theta_j, \log \alpha_r, \beta_r, d_{rm} \sim N(0.0, 1.0), \tag{4}$$

where $N(\mu, \sigma)$ indicates the normal distribution with mean μ and standard deviation σ. In the $EstIrtScore()$ function of Algorithm 1, we calculated the IRT scores through an EAP estimation using the Hermite–Gauss quadrature [101], which has been widely used in various IRT studies. Specifically, given rater parameter estimates, the score estimates are calculable as

$$\frac{\sum_{h=1}^{H} \theta'_h L(\boldsymbol{X}_j, \theta'_h) g(\theta'_h)}{\sum_{h=1}^{H} L(\boldsymbol{X}_j, \theta'_h) g(\theta'_h)}, \tag{5}$$

where θ'_h is the h-th integral point and H is the number of such points. We created the integration points by setting $H = 40$ and dividing the value range $[-4, 4]$ with an equal interval. In addition, $L(\boldsymbol{X}_j, \theta'_h)$ is the likelihood conditional on θ'_h for $\boldsymbol{X}_j$, which consists of observed scores for the j-th essay. $g(\theta'_h)$ indicates the prior probability for θ'_h. We assumed the standard normal distribution as the prior distribution.

We used a Tesla V100-SXM2 GPU to train DNN-AES models, whereas we used an Intel® Xeon® 2.00GHz CPU for training other AES models and score-integration methods, including the proposed method.

C. Results

Table III presents the experimental results, with bold text indicating maximum QWK values and underlined text representing the second-highest values for each prompt. In the table, the Avg. column shows the average QWK value for each method, and the p-value column shows the results of the one-tailed paired t-test between the proposed method using GMFRM and the other respective method.

According to Table III, the average QWK values of the individual AES models are around 0.7 in almost all models. Recent AES studies that used the ASAP dataset have generally reported average QWK values ranging from 0.7 to 0.8 [102], which are almost consistent with our results. Note that QWK values reported in different studies are not necessarily directly comparable, even when the same model and the same dataset are used, because they might employ different hyperparameter settings and methods for splitting data during cross-validation.

Comparison of the proposed method using GMFRM with the individual AES models shows that the proposed method is superior in all cases except for only one case (Hybrid–BERT in prompt 3) and shows a significantly higher average

TABLE IV
WAIC VALUES FOR THE GMFRM AND COMPARATIVE MODELS.

	Prompt							
	1	2	3	4	5	6	7	8
GMFRM	**3712.13**	**2328.44**	**2125.16**	**2998.14**	**2304.16**	**2894.49**	**8125.83**	**4785.87**
Consistency-fixed GMFRM	4688.65	2926.23	2717.16	3438.13	3122.17	3422.67	8766.69	5220.99
Severity-fixed GMFRM	3882.03	2508.65	2193.13	3279.71	2389.68	3005.90	8234.28	4792.55
Threshold-fixed GMFRM	4205.40	2641.49	2176.35	3195.11	2440.07	3086.20	8620.55	4832.36
MFRM	4944.37	3020.68	2730.38	3539.02	3142.40	3497.08	9152.08	5495.43

TABLE V
RATER PARAMETER ESTIMATES FOR PROMPT 1.

	α_r	β_r	d_{r2}	d_{r3}	d_{r4}	d_{r5}	d_{r6}	d_{r7}	d_{r8}	d_{r9}	d_{r10}	d_{r11}	d_{r12}
EASE (BLRR)	6.25	-0.83	0	-1.23	-1.66	-2.27	-1.60	-0.83	0.12	0.93	1.64	2.25	2.64
EASE (SVR)	2.73	-1.48	0	-1.27	-0.94	-0.79	-0.51	-0.18	0.16	0.34	0.75	1.12	1.31
XGBoost	2.39	-0.70	0	-2.00	-1.53	-1.47	-1.15	-0.82	-0.27	0.80	1.43	2.29	2.72
RNN	3.24	-0.48	0	-1.72	-1.58	-1.38	-1.43	-0.84	-0.54	0.30	1.43	2.76	2.98
SkipFlow	3.23	-0.72	0	-1.39	-2.20	-1.88	-1.09	-0.71	-0.13	0.71	1.35	1.92	3.42
Hybrid-BERT	6.72	-0.75	0	-1.48	-2.34	-1.40	-1.26	-0.76	-0.06	0.67	1.45	2.20	2.98
Human	1.74	-0.66	0	-0.82	-2.24	-1.17	-1.68	-0.48	-0.54	0.92	1.23	2.12	2.65

accuracy. The other score-integration methods also outperform the individual AES models in many cases, suggesting that integration of prediction scores from various AES models is effective.

Furthermore, compared with the conventional score-integration methods, the proposed method with GMFRM shows higher accuracy in almost all the cases and its average accuracy is significantly higher. This result indicates the effectiveness of the GMFRM-based score integration considering differences in characteristics of scoring behavior among the respective AES models. This result also suggests that the proposed method is expected to be effective with various datasets because the proposed method is superior in many prompts with different characteristics.

Among the proposed methods using different IRT models, the GMFRM provided the highest average accuracy at a significance level of 0.05. Thus, it would be reasonable to use the GMFRM in general. This result also suggests that all the rater parameters in the GMFRM are effective for improving the accuracy. To further examine the effectiveness of the GMFRM, we conducted a model comparison experiment using an information criterion. As the information criterion, we used the widely applicable information criterion (WAIC) [103], which is suitable for Bayesian estimation using MCMC. The WAIC was calculated for each cross-validation set, and these values were averaged for each prompt. Table IV shows the results. We highlighted the minimum scores in the table as bold text because the model minimizing the WAIC is regarded as the optimal model. According to the results, the GMFRM shows the best performance in all cases. The results suggest that the three rater characteristics (i.e., severity, consistency, and range restriction) vary among the AES models and the human rater, and thus the GMFRM is suitable compared to the other simpler models.

D. Analyzing the Characteristics of Scoring Behavior

Besides the improvement in scoring accuracy, a unique feature of the proposed method is its high interpretability, as explained in Section I. This subsection provides an interpretation of the scoring characteristics of each AES model based on the rater parameter values obtained from the GMFRM. As an example, Table V shows the rater parameter estimates of the AES models and the human rater for prompt 1. Note that, here, we estimated the GMFRM parameter using predicted scores of the AES models and the gold-standard human scores for all the essays for prompt 1. The AES prediction scores for all the essays can be obtained from the five-fold cross-validation explained in the previous section. Furthermore, based on the parameter values in Table V, we illustrate the IRCs for the AES models and the human rater in Figs. 4 and 5, respectively. In the figures, the horizontal axis shows the latent score θ_j, and the vertical axis shows the response probability for each category.

According to the table and figures, we can interpret the following characteristics:

- EASE (SVR) has an extremely low severity, reflecting the strong tendency to output the highest score ($k = 12$) overall.
- RNN and XGBoost show relatively low probabilities for categories 3, 4, and 5, suggesting the existence of a range restriction that avoids these categories. Moreover, XGBoost has another range restriction tendency to prefer category 8 slightly.
- EASE (BLRR), SkipFlow, and Hybrid–BERT have relatively high consistency. Furthermore, in the IRCs for these models, the curves for some categories [i.e., $k = 3$ and 4 in EASE (BLRR) and $k = 3$ in SkipFlow and Hybrid–BERT] are not displayed because these probabilities are extremely low, meaning that they have an extremely strong range restriction.
- The human rater tended to prefer categories 6, 8, and 10, and rarely used categories 3, 4, and 5, indicating the existence of a strong range restriction. As explained earlier, XGBoost shows a relatively similar range restriction, indicating that XGBoost imitates the human rater most precisely in prompt 1.
- Another interesting observation is that the AES models

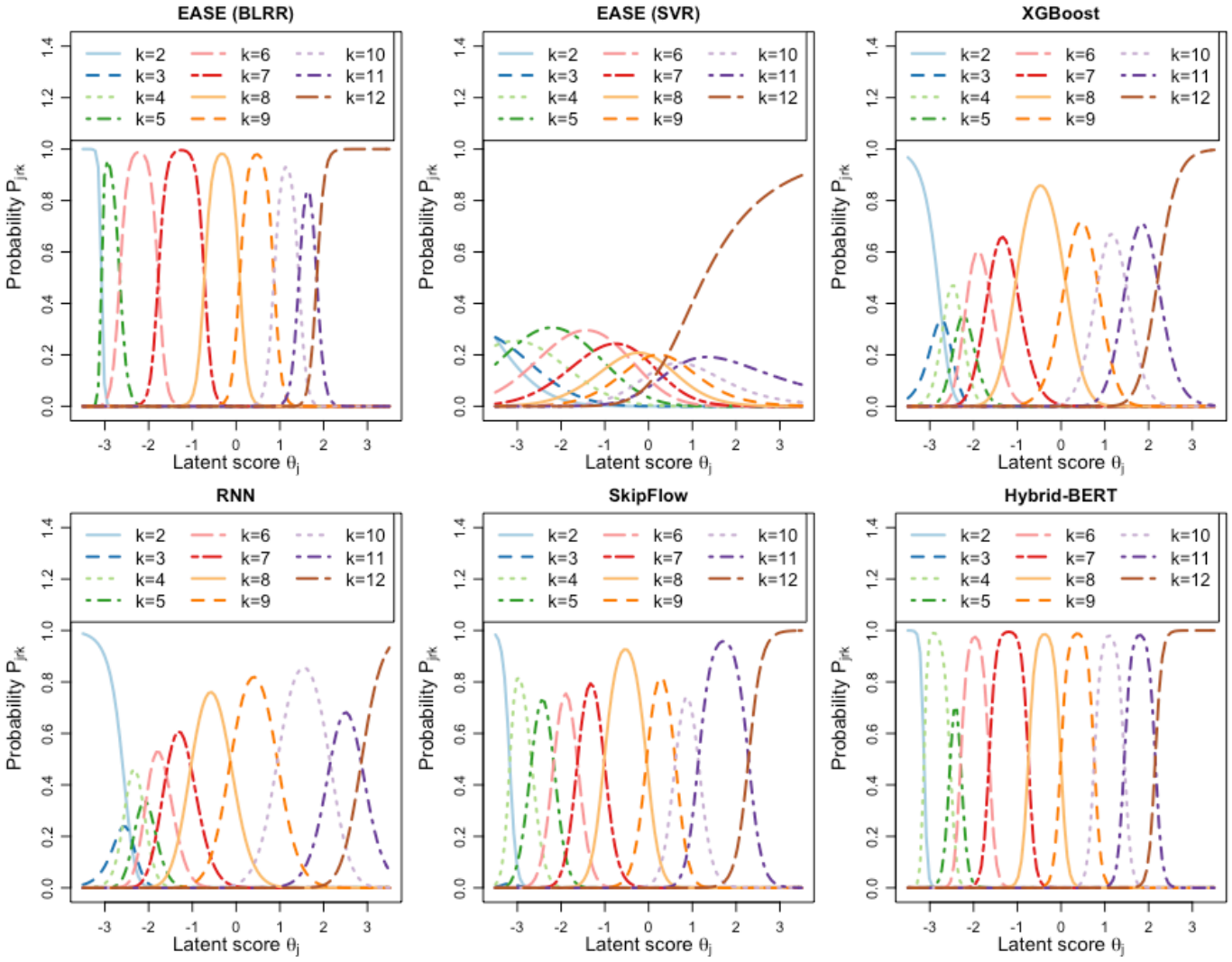

Fig. 4. IRCs of the AES models for prompt 1. Note that, in the EASE (BLRR), SkipFlow, and Hybrid–BERT models, the curves for some categories [$k = 3$ and 4 in EASE (BLRR), and $k = 3$ in SkipFlow and Hybrid-BERT] are not displayed because they have extremely small probabilities.

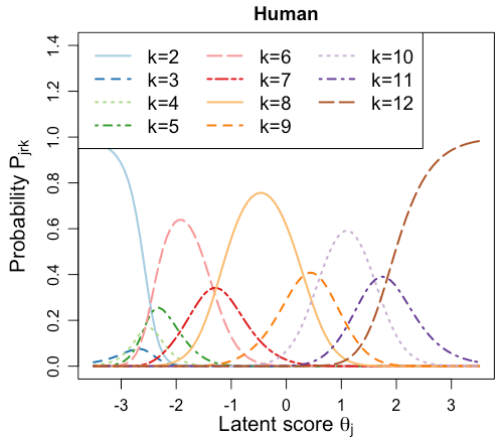

Fig. 5. IRC of the human rater for prompt 1.

show higher consistency than the human rater overall. One motivation of AES research is to realize consistent scoring, and this result demonstrates that AES can achieve it.

This analysis shows that the AES models have different scoring characteristics, indicating that integration of multiple AES models considering such characteristic differences is effective. Furthermore, the above discussion shows that the human rater who created the gold-standard scores used different scoring criteria compared with the AES models. This result indicates that the projection of the GMFRM-based latent scores θ_j into the human rater's rating scale is important to achieve high scoring accuracy.

E. Relation Between Predicted Scores of Individual AES Models and Integrated Scores of Proposed Method

To further examine the characteristics of individual AES models, this section describes the relations between the prediction scores of individual models and the integrated scores of the proposed method. Fig. 6 illustrates the relations in prompt 1. In each figure, the horizontal axis indicates the integrated scores of the proposed method using the GMFRM, and the vertical axis indicates the predicted scores of each AES model. The size of each bubble represents the appearance

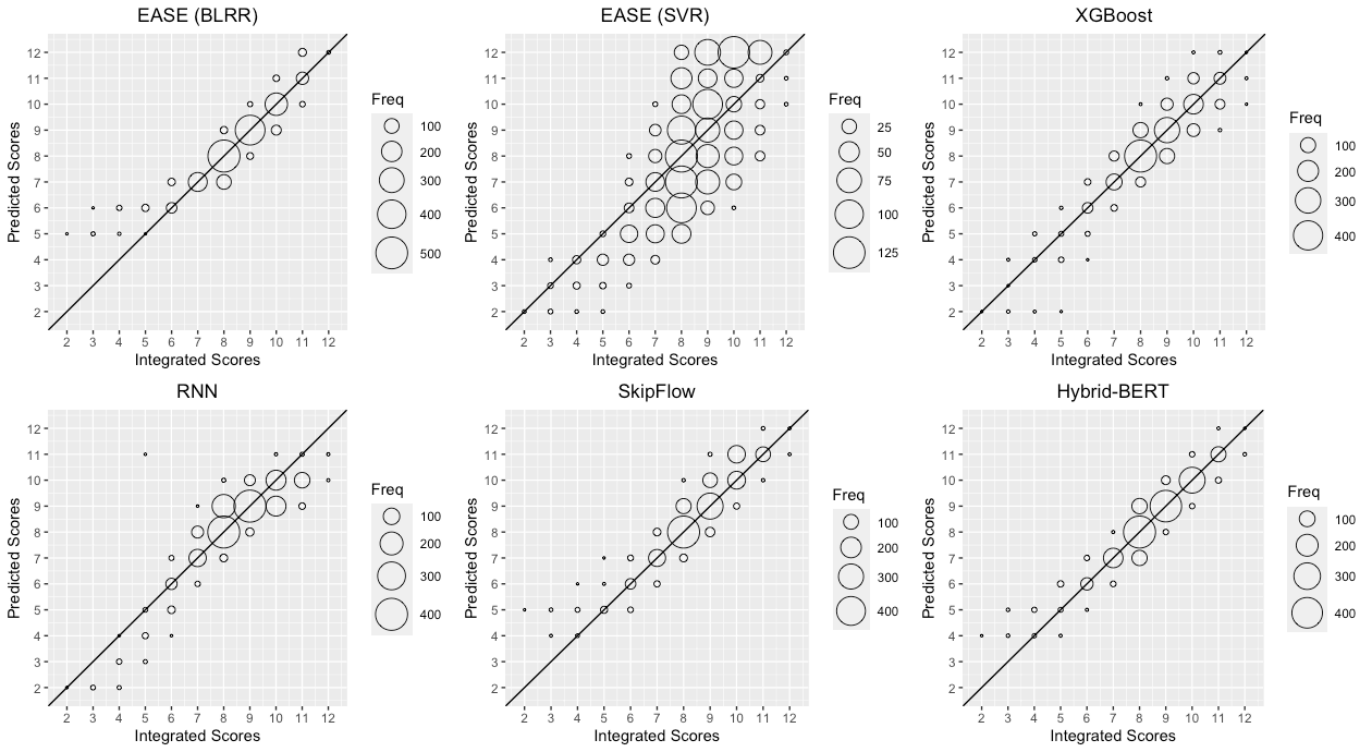

Fig. 6. Relation between predicted scores of individual AES and integrated scores of the proposed method in Prompt 1.

frequency of each data point, where a larger bubble represents a higher frequency.

According to Fig. 6, EASE (SVR) shows an extremely different tendency compared with the other models. Specifically, EASE (SVR) tends to overuse the high scores because it is extremely lenient, as described in the previous section. Also, EASE (SVR) cannot distinguish essays with medium or above qualities due to its extreme leniency. Thus, within the middle or above score range, its prediction scores substantially differ from the integrated scores.

EASE (BLRR), SkipFlow, and Hybrid–BERT tend to avoid several low-score categories, such as 2, 3, and 4, which is consistent with the fact that their IRCs represent extremely low probabilities for some of these categories, as explained above. The figures for these models also show that the proposed method can predict scores that some models do not produce at all.

XGBoost, which has characteristics most similar to those of a human rater, predicts scores that agree well with the integrated scores overall. The RNN also shows a relatively high agreement with the integrated scores. However, within the middle score range, that is, 7–9, the RNN shows a slightly larger disagreement with the integrated scores than XGBoost does. The reason is that XGBoost captured the tendency that the human rater prefers category 8, as explained in the previous section, whereas the RNN could not do that properly.

The above discussions demonstrate that the proposed method calculated the integrated scores while considering characteristics of scoring behavior in each AES model and their similarity to that of the human rater.

F. Relation Between Proposed Method Effectiveness and AES Characteristic Diversity

We can expect that the effectiveness of the proposed method will increase when the characteristic difference among the individual models becomes large. This subsection examines this hypothesis.

For this analysis, we quantified the characteristic differences among AES models using the mean absolute differences in IRCs, which have been used for IRT equating [104]. The difference metric for two AES models r and r' is defined as follows:

$$\delta(r,r') = \int \left| \sum_{k=1}^{K} kP_{jrk}(\theta) - \sum_{k=1}^{K} kP_{jr'k}(\theta) \right| d\theta \approx \frac{1}{H} \sum_{h=1}^{H} \left| \sum_{k=1}^{K} kP_{jrk}(\theta'_h) - \sum_{k=1}^{K} kP_{jr'k}(\theta'_h) \right|, \quad (6)$$

where $P_{jrk}(\theta)$ indicates the GMFRM-based probability calculated in (2) given the ability value θ, $\{\theta'_h | h \in \{1,\ldots,H\}\}$ is a collection of integration points, and H is the number of points. We created the integration points by setting $H = 40$ and dividing the value range $[-4,4]$ with an equal interval.

We calculated the distance metrics $\delta(r,r')$ for all the pairs of AES models and for all the pairs between the human rater and the AES models. The results are shown in Table VI.

First, focusing on the results for prompt 1, we can confirm that the metric between XGBoost and the human rater, which have similar characteristics of IRCs as explained earlier, shows a small value. Furthermore, the metric values between EASE (SVR), which has an extremely different IRC, as shown in

TABLE VI
IRC DIFFERENCE METRIC VALUES AMONG AES MODELS AND THOSE AMONG THE HUMAN RATER AND AES MODELS.

		Prompt							
		1	2	3	4	5	6	7	8
EASE (BLRR)	EASE (SVR)	0.794	0.571	0.443	0.458	0.403	0.406	1.287	2.627
EASE (BLRR)	XGBoost	0.604	0.550	0.324	0.289	0.360	0.407	0.989	2.044
EASE (BLRR)	RNN	0.789	0.648	0.287	0.322	0.311	0.416	1.977	2.183
EASE (BLRR)	SkipFlow	0.563	0.543	0.253	0.342	0.329	0.463	0.836	2.044
EASE (BLRR)	Hybrid-BERT	0.689	0.342	0.162	0.259	0.210	0.302	0.531	2.287
EASE (BLRR)	Human	0.595	0.562	0.400	0.320	0.387	0.499	1.232	2.156
EASE (SVR)	XGBoost	0.787	0.663	0.463	0.469	0.448	0.372	1.287	1.737
EASE (SVR)	RNN	0.893	0.752	0.470	0.525	0.442	0.308	2.159	1.770
EASE (SVR)	SkipFlow	0.667	0.634	0.391	0.496	0.370	0.327	1.248	1.742
EASE (SVR)	Hybrid-BERT	0.883	0.574	0.451	0.452	0.408	0.383	1.276	3.407
EASE (SVR)	Human	0.773	0.647	0.507	0.479	0.449	0.384	1.199	1.575
XGBoost	RNN	0.530	0.362	0.279	0.293	0.241	0.388	1.947	0.930
XGBoost	SkipFlow	0.495	0.330	0.335	0.300	0.304	0.441	0.851	0.631
XGBoost	Hybrid-BERT	0.705	0.464	0.337	0.256	0.368	0.327	0.841	3.025
XGBoost	Human	0.263	0.310	0.243	0.181	0.186	0.438	0.855	0.836
RNN	SkipFlow	0.693	0.415	0.272	0.210	0.256	0.235	1.819	0.992
RNN	Hybrid-BERT	0.861	0.577	0.257	0.377	0.322	0.398	1.929	3.124
RNN	Human	0.553	0.400	0.359	0.255	0.280	0.295	2.001	0.908
SkipFlow	Hybrid-BERT	0.661	0.435	0.252	0.383	0.331	0.450	0.684	3.024
SkipFlow	Human	0.493	0.347	0.399	0.270	0.295	0.245	0.987	0.951
Hybrid–BERT	Human	0.699	0.477	0.410	0.312	0.390	0.448	1.117	3.090
Average		0.728	0.524	0.332	0.362	0.340	0.375	1.311	2.104

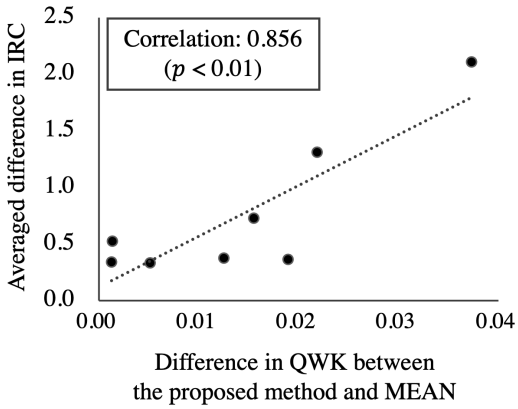

Fig. 7. Relation Between Characteristic Diversity Among AES Models and the Proposed Method Effectiveness.

Fig. 4, and the other models tend to be high. These results suggest that this metric reflects the scoring characteristic differences appropriately.

Next, focusing on the average row in Table VI, we can confirm that average metric values are relatively large in prompts 7 and 8. Furthermore, according to Table III, the proposed method using GMFRM shows large improvements in these two prompts compared with conventional integration methods, such as MEAN and VOTE. Here, Fig. 7 shows the relation between the average $\delta(r, r')$ values and the difference in the QWK values between the proposed method and the MEAN method. In the figure, the horizontal axis indicates the difference in the QWK values between the proposed method using GMFRM and the MEAN method, the vertical axis indicates the average values of the IRC difference metric $\delta(r, r')$, each plot indicates the results for a prompt, and the dotted line indicates the regression line. The figure shows a strong correlation. In particular, the Pearson correlation coefficient was 0.856, and it was significant ($p < 0.01$). A similar result was obtained between the proposed method and the VOTE method. Specifically, the correlation was 0.840, and it was also significant ($p < 0.01$). From these results, we can conclude that the effectiveness of the proposed method tends to increase with increasing differences between the characteristics of the scoring behavior among the AES models.

G. Relation Between Proposed Method Effectiveness and Prompt Characteristics

This subsection examines the relationship between the prompt characteristics and the proposed method effectiveness to investigate what prompt-dependent factors affect the performance of the proposed method. In this analysis, we regarded the difference in the QWK values between the proposed method and the MEAN method as the proposed method effectiveness, in the same way as the analysis of Section VI-F.

Fig. 8 shows the relation between the proposed method effectiveness and the prompt-dependent factors, namely, the average essay length, the score ranges, and the essay types. In the figure, the left panel shows the relation with the average essay length, the center panel shows that with the score range, and the right panel shows that with the essay type.

The figure shows that the effectiveness of the proposed method tends to increase with increasing the essay length and the score ranges. In addition, the effectiveness of the proposed method is relatively high for the narrative-type prompts compared to the other types. These results might indicate that these prompt-dependent factors affect the performance of the proposed method. However, it should be noted that, in the ASAP dataset, the essays for the narrative-type prompts are longer and have greater numbers of score categories than those for the other prompts, which might emphasize the characteristics difference among the individual models. As

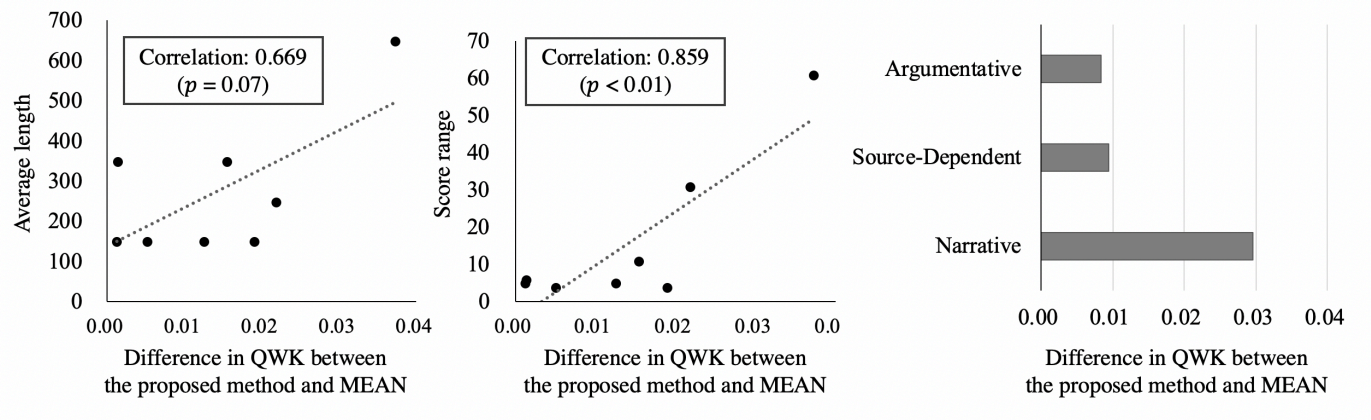

Fig. 8. Relation Between Prompt-Dependent Factors and the Proposed Method Effectiveness.

TABLE VII
COMPUTATIONAL TIMES FOR MODEL TRAINING (SECONDS).

		Prompt							
		1	2	3	4	5	6	7	8
Individual models	EASE (BLRR)	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1
	EASE (SVR)	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1
	XGBoost	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1
	RNN	55	58	28	27	31	32	34	30
	SkipFlow	171	193	70	76	81	85	119	96
	Hybrid-BERT	452	445	445	417	441	458	417	218
	Total	679	697	543	521	554	575	572	345
Conventional integration models	Stacking (Linear)	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1
	Stacking (Ridge)	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1
	Stacking (SVR)	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1
	Stacking (Boosting)	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1
Proposed method	GMFRM	1132	266	292	123	306	298	1255	688
	Consistency-fixed GMFRM	416	157	149	114	152	175	882	712
	Severity-fixed GMFRM	1134	148	131	116	157	148	1199	704
	Threshold-fixed GMFRM	475	173	223	126	252	209	702	501
	MFRM	311	141	123	114	153	141	520	499

discussed in Section VI-F, the characteristics difference among the individual models affects the effectiveness of the proposed method, meaning that these prompt-dependent factors might have no or small direct impact on increasing the effectiveness of the proposed method. A further analysis based on large-scale experiments with various datasets is required to investigate the factors affecting the effectiveness of the proposed method in more detail, and this task remains as future work.

H. Computational Costs

This subsection investigates the computational cost for each method. Specifically, we calculated the time for training each AES model and the score-integration models. We calculated the times for each partition in the five-fold cross-validation.

Table VII shows the average times. Here, the time for data preprocessing and computing manually designed features was excluded. According to Table VII, the feature-engineering approach models can be trained in less than 1 second, whereas DNN-AES models take a few minutes even though they use a GPU. The total time to complete the training of all individual AES models is about 5 to 10 minutes.

The conventional score-integration methods can also be trained in less than 1 second. However, the proposed method requires up to about 20 minutes, which is a relatively long time compared with the others. The main reason is that we used the MCMC for the IRT parameter estimation because it is expected to provide high estimation accuracy. If faster estimation is required, other estimation algorithms are available, including the marginal maximum likelihood estimation and the maximum a posteriori estimation using the Newton–Raphson method and EAP estimation with variational Bayesian methods. However, the total training time for the proposed method, including the time to train individual AES models, is about 30 minutes at most, which will generally be acceptable in practical use.

We also computed the time for scoring a new essay using each trained model. Consequently, the time to compute the score of a single essay was less than 0.1 seconds in all models, which is also sufficient for practical applications.

VII. CONCLUSION

In this study, we proposed a method that uses IRT to integrate prediction scores from various AES models while taking into account differences in scoring behavior characteristics. Specifically, we proposed the use of IRT incorporating rater characteristic parameters by regarding AES models as raters. We performed experiments with a benchmark dataset to demonstrate that the proposed method with the latest IRT

model, GMFRM, provided significantly higher accuracy than individual AES models and conventional integration methods. We also showed that the scoring characteristics could be interpreted for each AES model based on the IRT parameters, and confirmed a large characteristic variety among AES models and between a human rater and AES models. Furthermore, we demonstrated that the effectiveness of the proposed method tends to increase as this characteristic variety increases.

VIII. Limitations and future work

We will evaluate the effectiveness of the proposed method by adding more distinctive models because our method is expected to improve accuracy with the addition of various AES models, as demonstrated in Section VI-E. Furthermore, although this study employed multiple models with entirely different architectures as base models, *input manipulation*, an ensemble learning method in which multiple models are constructed using different training subsets, as in AdaBoost and Bagging, may also be suitable for preparing multiple different models. Examining the proposed method using an input manipulation method is our future work. Another extension of the proposed method based on other meta-learning methods, such as *mixture of experts*, may also be a possible future direction. We also need a further analysis based on large-scale experiments with various datasets to elucidate the factors affecting the effectiveness of the proposed method, as discussed in Section VI-G.

As shown in Section VI-F, the proposed method provides information representing the characteristics of scoring behavior for AES models. Such information would be helpful not only to understand the scoring characteristics of each model but also to consider an improvement or extension of each model. We thus plan to examine how individual AES models can be improved based on the information obtained from the proposed method.

Another future direction is to consider the use of the proposed method for enhancing collaboration between AES models and human raters because the use of AES to support human raters is also a recent popular research topic [105], [106].

Moreover, in this study, we assumed that the gold-standard scores in training data are given by a single human rater. These scores, however, are often created by aggregating multiple scores given by multiple human raters, as pointed out in some previous studies [61], [63], [66]. The proposed method can be easily applied to data in which each essay has multiple human rater scores. In future studies, we plan to apply the proposed method to such data and evaluate its effectiveness.

This study focused on a *prompt-specific scoring task*, the most common AES task, in which an AES model is trained for a prompt and the trained model is used to evaluate essays for the same prompt. Another important AES task is a *cross-prompt scoring task*, in which no or few training data for a target prompt exist, but data for other prompts are available. Cross-prompt scorings are often realized using domain adaptation or transfer learning techniques, which are studied widely in AI and machine learning domains. However, the number of papers dealing with this task remains limited [13]. We will examine an extension of the proposed method for such tasks in future work.

Another future direction relates to the use of the IRT. An extension of the proposed method using cognitive diagnosis models (CDMs), including DINA (deterministic inputs, noisy and gate) [107] and NeuralCD (neural cognitive diagnosis) [108] models, might contribute to improving the interpretability of the attributes (e.g., knowledge or skills) measured by the target essay writing test. However, most CDMs cannot be directly applied to our framework because they require a Q-matrix, which defines the required attributes for each test item, and it is not generally included in most existing datasets for AES. We would like to investigate the possibility of integrating CDMs into our framework in the future.

Furthermore, analyzing the differences in the item characteristics between the essay writing test items and the other types of items, such as multiple choice questions, is an important issue in the field of educational measurement, although this is outside the scope of this study. A fusion of IRT and AES, as in the proposed method, might be helpful for realizing a detailed analysis of this aspect, which is also a future work. In addition, although this study focused on the AES context, applying the proposed method to other text-scoring tasks, including ASAG, is also future work.

Appendix

This appendix describes the detailed architectures of the DNN-based automatic feature extraction approach models and the hybrid model, which were used in our experiments.

A. CNN-RNN-Based Model

Fig. 9 shows the architecture of the CNN-RNN-based model [27]. This model calculates a score for a targeted essay, which is defined as a sequence of words $\{w_1, \ldots, w_n\}$ (where w_t is the t-th word in the essay and n is the number of words), through five DNN layers.

1) The first layer is the lookup table layer that transforms each word into a D-dimensional word-embedding representation, in which words with similar meanings have similar vector representations.
2) The second layer, the convolution layer, uses a CNN to extract N-gram-level features from the sequence of word-embedding vectors. In this layer, each word vector is transformed to another vector representation that reflects dependencies among N-adjacent words. This layer is often omitted in some extension models.
3) The third layer, the recurrent layer, uses an RNN to transform each output vector from the convolution layer into another vector representation that reflects the context of the essay. A long short-term memory network is generally used as the RNN.
4) The fourth layer is a pooling layer that transforms the output vector sequence of the recurrent layer into an aggregated fixed-length hidden vector by averaging the vector sequence.

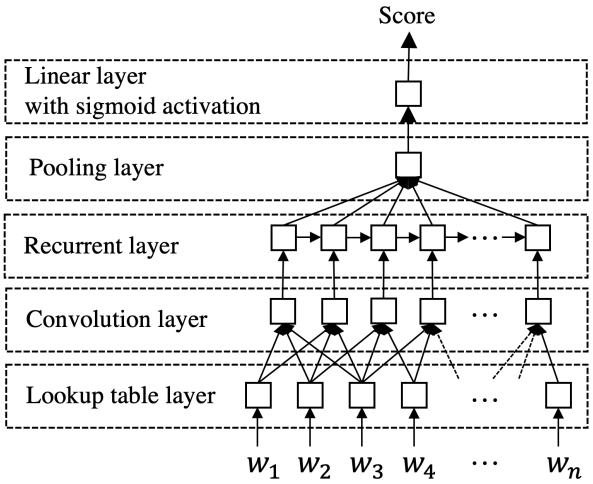

Fig. 9. Architecture of CNN-RNN-based model.

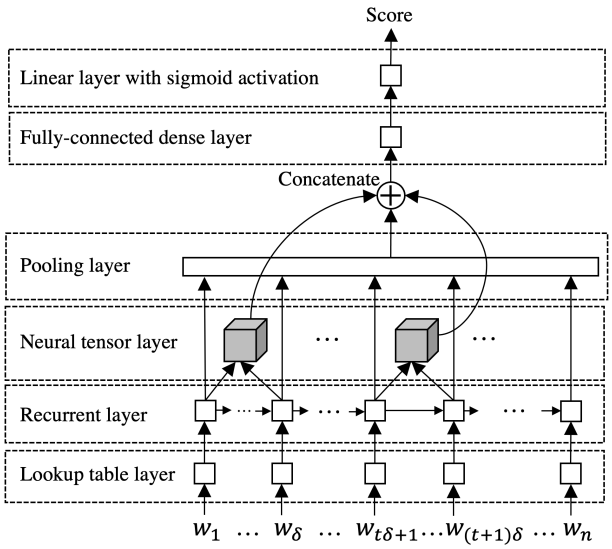

Fig. 10. Architecture of SkipFlow model (δ is a hyperparameter that controls the relevance width).

5) The last layer, the linear layer with sigmoid activation, projects the output vector of the pooling layer to a scalar value by using a sigmoid function.

B. SkipFlow Model

The SkipFlow model [32] is a representative DNN-AES model with a function that captures textual coherence directly, as explained in Section III-B. The architecture of this model is shown in Fig. 10. The SkipFlow model consists of almost the same components as those used in the CNN-RNN-based model. Specifically, the model uses the lookup table layer, the recurrent layer, the pooling layer, and the linear layer with sigmoid activation. The main difference is the incorporation of the neural tensor layer. The neural tensor layer takes two positional outputs of the recurrent layer that are collected from different time steps and computes the similarity between each pair of positional outputs. The similarity score is regarded as a neural coherence feature between the two selected positions. The list of the similarity scores is concatenated with the pooling layer output, and the concatenated vector is used to predict an essay score.

C. BERT-based Hybrid AES Model

Fig. 11 shows the architecture of the BERT-based hybrid AES model proposed in [42]. This model concatenates manually designed essay-level features to the distributed essay representation, which is the input vector for the last linear layer in the BERT-based AES model.

Note that, as explained in Section III-B, the BERT is a transformer-based model pre-trained on massive amounts of unlabeled text data for two tasks, namely, *masked language modeling* and *next-sentence prediction*. Masked language modeling involves predicting the words that are masked out of the input text, whereas the next-sequence prediction involves predicting whether two given sentences are adjacent.

AES using the pre-trained BERT can be realized by the following procedure:

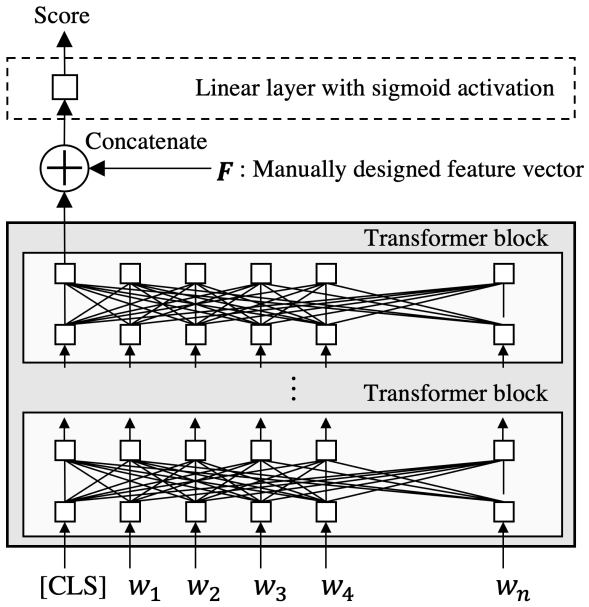

Fig. 11. Architecture of BERT-based hybrid model ($\boldsymbol{F}$ indicates a manually designed feature vector).

1) Add a special classification ([CLS]) token to the beginning of each essay text.
2) Add a linear layer with sigmoid activation over the output corresponding to this token because the BERT output for this token is known to be a distributed representation for specific input text.
3) Fine-tune this BERT-based AES model using a training dataset that consists of essays and corresponding scores.

To implement the BERT-based hybrid AES model, manually designed essay-level features are concatenated with the BERT output for the [CLS] token in step 2 of the above procedure.

REFERENCES

[1] H. J. Bernardin, S. Thomason, M. R. Buckley, and J. S. Kane, "Rater rating-level bias and accuracy in performance appraisals: The impact of rater personality, performance management competence, and rater accountability," *Human Resour. Manage.*, vol. 55, no. 2, pp. 321–340, Mar.–Apr. 2016.

[2] O. L. Liu, L. Frankel, and K. C. Roohr, "Assessing critical thinking in higher education: Current state and directions for next-generation assessment," *ETS Res. Rep. Ser.*, vol. 2014, no. 1, pp. 1–23, June 2014.

[3] Y. Rosen and M. Tager, "Making student thinking visible through a concept map in computer-based assessment of critical thinking," *J. Educ. Comput. Res.*, vol. 50, no. 2, pp. 249–270, Mar. 2014.

[4] Y. Abosalem, "Assessment techniques and students' higher-order thinking skills," *Int. J. Secondary Educ.*, vol. 4, no. 1, pp. 1–11, Feb. 2016.

[5] M. A. Hussein, H. A. Hassan, and M. Nassef, "Automated language essay scoring systems: A literature review," *PeerJ Comput. Sci.*, vol. 5, p. e208, Aug. 2019.

[6] Z. Ke and V. Ng, "Automated essay scoring: A survey of the state of the art," in *Proc. 28th Int. Joint Conf. Artificial Intelligence*, Macao, China, Aug. 10–16 2019, pp. 6300–6308.

[7] J. Wang and M. S. Brown, "Automated essay scoring versus human scoring: A correlational study," *Contemporary Issues Technol. Teacher Educ.*, vol. 8, no. 4, pp. 310–325, 2008.

[8] C. Lu and M. Cutumisu, "Integrating deep learning into an automated feedback generation system for automated essay scoring," in *Proc. Int. Conf. Educational Data Mining*, 2021.

[9] M. Liu, Y. Li, W. Xu, and L. Liu, "Automated essay feedback generation and its impact on revision," *IEEE Trans. Learn. Technol*, vol. 10, no. 4, pp. 502–513, 2017.

[10] R. Ridley, L. He, X. yu Dai, S. Huang, and J. Chen, "Automated cross-prompt scoring of essay traits," in *Proc. 35th. AAAI Conf. Artificial Intelligence*, vol. 35, no. 15, Online, Feb. 1–9 2021, pp. 13 745–13 753.

[11] T. Shibata and M. Uto, "Analytic automated essay scoring based on deep neural networks integrating multidimensional item response theory," in *Proc. Int. Conf. Computational Linguistics*, Gyeongju, Republic of Korea, Oct. 2022, pp. 2917–2926.

[12] Y. He, F. Jiang, X. Chu, and P. Li, "Automated Chinese essay scoring from multiple traits," in *Proc. Int. Conf. Computational Linguistics*, Gyeongju, Republic of Korea, Oct. 2022, pp. 3007–3016.

[13] M. Uto, "A review of deep-neural automated essay scoring models," *Behaviormetrika*, vol. 48, no. 2, pp. 4459–484, July 2021.

[14] J. Burstein, "The e-rater® scoring engine: Automated essay scoring with natural language processing," in *Automated essay scoring: A cross-disciplinary perspective*, M. D. Shermis and J. Burstein, Eds. Mahwah, NJ, USA: Lawrence Erlbaum Associates Publishers, 2003, pp. 113–121.

[15] Y. Attali and J. Burstein, "Automated essay scoring with e-rater v.2," *J. Technol. Learn. Assessment*, vol. 4, no. 3, pp. 1–31, Feb. 2006.

[16] P. Phandi, K. M. A. Chai, and H. T. Ng, "Flexible domain adaptation for automated essay scoring using correlated linear regression," in *Proc. 2015 Conf. Empirical Methods in Natural Language Processing*, Lisbon, Portugal, Sep. 17–21 2015, pp. 431–439.

[17] B. Beigman Klebanov, M. Flor, and B. Gyawali, "Topicality-based indices for essay scoring," in *Proc. 11th Workshop on Innovative Use of NLP for Building Educational Applications*, San Diego, CA, USA, June 16–17 2016, pp. 63–72.

[18] M. Cozma, A. Butnaru, and R. T. Ionescu, "Automated essay scoring with string kernels and word embeddings," in *Proc. 56th Annu. Meeting Association for Computational Linguistics*, Melbourne, Australia, July 15-20 2018, pp. 503–509.

[19] H. V. Nguyen and D. J. Litman, "Argument mining for improving the automated scoring of persuasive essays," in *Proc. 32nd AAAI Conf. Artificial Intelligence*, vol. 32, no. 1, New Orleans, Lousiana, USA, Feb. 2–7 2018, pp. 5892–5899.

[20] H. K. Janda, A. Pawar, S. Du, and V. Mago, "Syntactic, semantic and sentiment analysis: The joint effect on automated essay evaluation," *IEEE Access*, vol. 7, pp. 108 486–108 503, 2019.

[21] Y. Yang, L. Xia, and Q. Zhao, "An automated grader for Chinese essay combining shallow and deep semantic attributes," *IEEE Access*, vol. 7, pp. 176 306–176 316, 2019.

[22] E. Amorim, M. Cançado, and A. Veloso, "Automated essay scoring in the presence of biased ratings," in *Proc. 2018 Conf. North American Chapter of Association for Computational Linguistics*, New Orleans, Louisiana, USA, June 1–6 2018, pp. 229–237.

[23] M. Dascalu, W. Westera, S. Ruseti, S. Trausan-Matu, and H. Kurvers, "ReaderBench learns dutch: Building a comprehensive automated essay scoring system for Dutch language," in *Proc. 18th Int. Conf. Artificial Intelligence in Education*, Wuhan, China, June 28–July 2 2017, pp. 52–63.

[24] M. D. Shermis and J. C. Burstein, *Automated Essay Scoring: A Cross-disciplinary Perspective*. Evanston, IL, USA: Routledge, 2002.

[25] C. Jin, B. He, K. Hui, and L. Sun, "TDNN: A two-stage deep neural network for prompt-independent automated essay scoring," in *Proc. 56th Annu. Meeting Association for Computational Linguistics*, Melbourne, Australia, July 15-20 2018, pp. 1088–1097.

[26] S. Yuan, T. He, H. Huang, R. Hou, and M. Wang, "Automated Chinese essay scoring based on deep learning," *Comput., Mater. & Continua*, vol. 65, no. 1, pp. 817–833, 2020.

[27] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring," in *Proc. 2016 Conf. Empirical Methods in Natural Language Processing*, Austin, Texas, USA, Nov. 1-5 2016, pp. 1882–1891.

[28] D. Alikaniotis, H. Yannakoudakis, and M. Rei, "Automatic text scoring using neural networks," in *Proc. 54th Annu. Meeting Association for Computational Linguistics*, Berlin, Germany, Aug. 7–12 2016, pp. 715–725.

[29] F. Dong and Y. Zhang, "Automatic features for essay scoring – an empirical study," in *Proc. 2016 Conf. Empirical Methods in Natural Language Processing*, Austin, Texas, USA, Nov. 1–5 2016, pp. 1072–1077.

[30] F. Dong, Y. Zhang, and J. Yang, "Attention-based recurrent convolutional neural network for automatic essay scoring," in *Proc. 21st Conf. Computational Natural Language Learning*, Vancouver, Canada, Aug. 3–4 2017, pp. 153–162.

[31] Y. Farag, H. Yannakoudakis, and T. Briscoe, "Neural automated essay scoring and coherence modeling for adversarially crafted input," in *Proc. 2018 Conf. North American Chapter of Association for Computational Linguistics*, New Orleans, Louisiana, USA, June 1–6 2018, pp. 263–271.

[32] Y. Tay, M. Phan, A. T. Luu, and S. C. Hui, "SkipFlow: Incorporating neural coherence features for end-to-end automatic text scoring," in *Proc. 32nd AAAI Conf. Artificial Intelligence*, vol. 32, no. 1, Feb. 2–7 2018, pp. 5948–5955.

[33] M. Mesgar and M. Strube, "A neural local coherence model for text quality assessment," in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing*, Brussels, Belgium, Oct. 31–Nov. 4 2018, pp. 4328–4339.

[34] Y. Wang, Z. Wei, Y. Zhou, and X. Huang, "Automatic essay scoring incorporating rating schema via reinforcement learning," in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing*, Brussels, Belgium, Oct. 31–Nov. 4 2018, pp. 791–797.

[35] F. S. Mim, N. Inoue, P. Reisert, H. Ouchi, and K. Inui, "Unsupervised learning of discourse-aware text representation for essay scoring," in *Proc. 57th Annu. Meeting Association for Computational Linguistics, Student Research Workshop*, Florence, Italy, July 28–Aug. 2 2019, pp. 378–385.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 4–7 2017, pp. 5998–6008.

[37] F. Nadeem, H. Nguyen, Y. Liu, and M. Ostendorf, "Automated essay scoring with discourse-aware neural models," in *Proc. 14th Workshop on Innovative Use of NLP for Building Educational Applications*, Florence, Italy, Aug. 2 2019, pp. 484–493.

[38] P. U. Rodriguez, A. Jafari, and C. M. Ormerod, "Language models and automated essay scoring," *arXiv*, Sep. 2019.

[39] R. Yang, J. Cao, Z. Wen, Y. Wu, and X. He, "Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking," in *Findings of the Association for Computational Linguistics*, Online, Nov. 16–20 2020, pp. 1560–1569.

[40] E. Mayfield and A. W. Black, "Should you fine-tune BERT for automated essay scoring?" in *Proc. 15th Workshop on Innovative Use of NLP for Building Educational Applications*, 2020, pp. 151–162.

[41] J. Xue, X. Tang, and L. Zheng, "A hierarchical BERT-based transfer learning approach for multi-dimensional essay scoring," *IEEE Access*, vol. 9, pp. 125 403–125 415, 2021.

[42] M. Uto, Y. Xie, and M. Ueno, "Neural automated essay scoring incorporating handcrafted features," in *Proc. 28th Int. Conf. Computational Linguistics*, Barcelona, Spain (Online), Dec. 8-13 2020, pp. 6077–6088.

[43] X. Li, M. Chen, J. Nie, Z. Liu, Z. Feng, and Y. Cai, "Coherence-based automated essay scoring using self-attention," in *Proc. Chinese*

Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Springer International Publishing, 2018, pp. 386–397.

[44] X. Li, H. Yang, S. Hu, J. Geng, K. Lin, and Y. Li, "Enhanced hybrid neural network for automated essay scoring," *Expert Syst.*, 2022.

[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 17th Annu. Conf. North American Chapter of Association for Computational Linguistics : Human Language Technologies*, Minneapolis, MN, USA, June 2–7 2019, pp. 4171–4186.

[46] T. Dasgupta, A. Naskar, L. Dey, and R. Saha, "Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring," in *Proc. 5th Workshop on Natural Language Processing Techniques for Educational Applications*, Melbourne, Australia, July 19 2018, pp. 93–102.

[47] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining and Knowl. Discov.*, vol. 8, no. 4, p. e1249, 2018.

[48] T. Eckes, *Introduction to Many-Facet Rasch Measurement: Analyzing and Evaluating Rater-Mediated Assessments*. Bern, Switzerland: Peter Lang Pub. Inc., 2011.

[49] M. Uto and M. Ueno, "Empirical comparison of item response theory models with rater's parameters," *Heliyon*, vol. 4, no. 5, pp. 1–32, May 2018.

[50] R. J. Patz and B. Junker, "Applications and extensions of MCMC in IRT: Multiple item types, missing data, and rated responses," *J. Educ. Behav. Statist.*, vol. 24, no. 4, pp. 342–366, Dec. 1999.

[51] J. M. Linacre, *Many-faceted Rasch Measurement*. Chicago, IL, USA: MESA Press, 1989.

[52] R. J. Patz, B. W. Junker, M. S. Johnson, and L. T. Mariano, "The hierarchical rater model for rated test items and its application to large-scale educational assessment data," *J. Educ. Behav. Statist.*, vol. 27, no. 4, pp. 341–384, Dec. 2002.

[53] L. T. DeCarlo, Y. K. Kim, and M. S. Johnson, "A hierarchical rater model for constructed responses, with a signal detection rater model," *J. Educational Meas.*, vol. 48, no. 3, pp. 333–356, Sep. 2011.

[54] M. Wilson and M. Hoskens, "The rater bundle model," *J. Educational Behavioral Statist.*, vol. 26, no. 3, pp. 283–306, Sep. 2001.

[55] H. J. Shin, S. Rabe-Hesketh, and M. Wilson, "Trifactor models for Multiple-Ratings data," *Multivariate Behav. Res.*, vol. 54, no. 3, pp. 360–381, Mar. 2019.

[56] M. Uto and M. Ueno, "Item response theory for peer assessment," *IEEE Trans. Learn. Technol*, vol. 9, no. 2, pp. 157–170, Apr.–Jun. 2016.

[57] ——, "A generalized many-facet Rasch model and its Bayesian estimation using Hamiltonian Monte Carlo," *Behaviormetrika*, vol. 47, no. 2, pp. 469–496, May 2020.

[58] M. Uto, "Rater-effect IRT model integrating supervised LDA for accurate measurement of essay writing ability," in *Proc. 20th Int. Conf. Artificial Intelligence in Education*, Chicago, IL, USA, June 25–29 2019, pp. 494–506.

[59] M. Uto, N. Duc Thien, and M. Ueno, "Group optimization to maximize peer assessment accuracy using item response theory and integer programming," *IEEE Trans. Learn. Technol*, vol. 13, no. 1, pp. 91–106, Feb. 2020.

[60] T. Nguyen, M. Uto, Y. Abe, and M. Ueno, "Reliable peer assessment for team project based learning using item response theory," in *Proc. 23rd Int. Conf. Comput. Educ.*, Hangzhou, China, Nov. 30–Dec. 4 2015, pp. 144–153.

[61] M. Uto and M. Okano, "Robust neural automated essay scoring using item response theory," in *Proc. 21st Int. Conf. Artificial Intelligence in Education*, Online, July. 6–10 2020, pp. 549–561.

[62] ——, "Learning automated essay scoring models using item-response-theory-based scores to decrease effects of rater biases," *IEEE Trans. Learn. Technol*, vol. 14, no. 6, pp. 763–776, Jan. 2022.

[63] S. A. Wind, E. W. Wolfe, G. E. Jr., P. Foltz, and M. Rosenstein, "The influence of rater effects in training sets on the psychometric quality of automated scoring for writing assessments," *Int. J. Testing*, vol. 18, no. 1, pp. 27–49, Nov. 2018.

[64] S. Li, S. Ge, Y. Hua, C. Zhang, H. Wen, T. Liu, , and W. Wang, "Coupled-view deep classifier learning from multiple noisy annotators," in *Proc. 34th AAAI Conf. Artificial Intelligence*, vol. 34, no. 4, New York, NY, USA, Feb. 7–12 2020, pp. 4667–4674.

[65] J. Amidei, P. Piwek, and A. Willis, "Identifying annotator bias: A new IRT-based method for bias identification," in *Proc. 28th Computational Linguistics*, Barcelona, Spain (Online), Dec. 8-13 2020, pp. 4787–4797.

[66] K. Zupanc and Z. Bosnić, "Increasing accuracy of automated essay grading by grouping similar graders," in *Proc. 8th Int. Conf. Web Intelligence, Mining and Semantics*, no. 35, Novi Sad, Serbia, June 25–27 2018, pp. 1–6.

[67] S. Bonthu, S. R. Sree, and M. K. Prasad, "Automated short answer grading using deep learning: A survey," in *Proc. Machine Learning and Knowledge Extraction*, 2021, pp. 61–78.

[68] S. Burrows, I. Gurevych, and B. Stein, "The eras and trends of automatic short answer grading," *Int. J. Artif. Intell. Educ.*, vol. 25, no. 1, pp. 60–117, 2015.

[69] C. Leacock and M. C. n, "C-rater: Automated scoring of short-answer questions," vol. 37, pp. 389–405, 2003.

[70] B. Riordan, A. Horbach, A. Cahill, T. Zesch, and C. M. Lee, "Investigating neural architectures for short answer scoring," in *Proc. Workshop on Innovative Use of NLP for Building Educational Applications*, 2017, pp. 159–168.

[71] T. Liu, W. Ding, Z. Wang, J. Tang, G. Y. Huang, and Z. Liu, "Automatic short answer grading via multiway attention networks," in *Proc. 20th Int. Conf. Artificial Intelligence in Education*, Chicago, IL, USA, June 25–29 2019, pp. 169–173.

[72] J. Lun, J. Zhu, Y. Tang, and M. Yang, "Multiple data augmentation strategies for improving performance on automatic short answer scoring," in *Proc. 34th AAAI Conf. Artificial Intelligence*, vol. 34, no. 9, New York, NY, USA, Feb. 7–12 2020, pp. 13 389–13 396.

[73] C. Sung, T. I. Dhamecha, and N. Mukhi, "Improving short answer grading using transformer-based pre-training," in *Proc. 20th Int. Conf. Artificial Intelligence in Education*, Chicago, IL, USA, June 25–29 2019, pp. 469–481.

[74] X. Gong and Z. A.-K. M. Al-Jepoori, "An attention-based deep model for automatic short answer score," in *Int. J. Comput. Sci. Softw. Eng.*, 2019, pp. 127–132.

[75] D. Gautam and V. Rus, "Using neural tensor networks for open ended short answer assessment," in *Proc. Int. Conf. Artificial Intelligence in Education*, 2020, pp. 191–203.

[76] Z. Li, Y. Tomar, and R. J. Passonneau, "A semantic feature-wise transformation relation network for automatic short answer grading," in *Proc. Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 6030–6040.

[77] J. Xie, K. Cai, L. Kong, J. Zhou, and W. Qu, "Automated essay scoring via pairwise contrastive regression," in *Proc. Int. Conf. Computational Linguistics*, Gyeongju, Republic of Korea, Oct. 2022, pp. 2724–2733.

[78] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," 2019.

[79] Ø. E. Andersen, R. Watson, Z. Yuan, and K. Y. F. Cheung, "Benefits of alternative evaluation methods for automated essay scoring," in *Proc. Int. Conf. Educational Data Mining*, 2021.

[80] F. Lord, *Applications of item response theory to practical testing problems*. Evanston, IL, USA: Routledge, 1980.

[81] F. Samejima, "Estimation of latent ability using a response pattern of graded scores," *Psychometrika Monography*, vol. 17, pp. 1–100, June 1969.

[82] E. Muraki, "A generalized partial credit model," in *Handbook of Modern Item Response Theory*, W. J. van der Linden and R. K. Hambleton, Eds. New York, NY, USA: Springer, 1997, pp. 153–164.

[83] M. J. Kolen and R. L. Brennan, *Test Equating, Scaling, and Linking*. Springer, 2014.

[84] M. Uto, "Accuracy of performance-test linking based on a many-facet Rasch model," *Behav. Res. Methods*, vol. 53, no. 4, pp. 1440–1454, 2021.

[85] N. L. A. Kassim, "Judging behaviour and rater errors: An application of the many-facet Rasch model," *GEMA Online J. Lang. Stud.*, vol. 11, no. 3, pp. 179–197, Sep. 2011.

[86] C. M. Myford and E. W. Wolfe, "Understanding Rasch measurement: Detecting and measuring rater effects using many-facet Rasch measurement: Part I," *J. Appl. Meas.*, vol. 4, no. 4, pp. 386–422, 2003.

[87] ——, "Understanding Rasch measurement: Detecting and measuring rater effects using many-facet Rasch measurement: Part II," *J. Appl. Meas.*, vol. 5, no. 2, pp. 189–227, 2004.

[88] F. Saal, R. Downey, and M. Lahey, "Rating the ratings: Assessing the psychometric quality of rating data," *Psychol. Bull.*, vol. 88, no. 2, pp. 413–428, Sep. 1980.

[89] T. Eckes, "Examining rater effects in TestDaF writing and speaking performance assessments: A many-facet Rasch analysis," *Lang. Assessment Quart.*, vol. 2, no. 3, pp. 197–221, Nov. 2005.

[90] G. J. Bean and N. K. Bowen, "Item response theory and confirmatory factor analysis: Complementary approaches for scale development," *J. Evidence-Based Social Work*, vol. 18, no. 6, pp. 597–618, 2021.

[91] R. J. Wirth and M. C. Edwards, "Item factor analysis: Current approaches and future directions," *Psychol Methods*, vol. 12, no. 1, pp. 58–79, 2008.
[92] S. Mathias and P. Bhattacharyya, "ASAP++: Enriching the ASAP automated essay grading dataset with essay attribute scores," in *Proc. Int. Conf. Language Resources Evaluation*, 2018, pp. 1169–1173.
[93] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.
[94] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting the Association for Computational Linguistics*, Baltimore, Maryland, June, 23–25 2014, pp. 55–60.
[95] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, Aug. 13–17 2016, pp. 785–794.
[96] J.-P. Fox, *Bayesian item response modeling: Theory and applications*. New York, NY, USA: Springer, 2010.
[97] M. D. Hoffman and A. Gelman, "The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1593–1623, Jan. 2014.
[98] S. Brooks, A. Gelman, G. Jones, and X. Meng, *Handbook of Markov Chain Monte Carlo*. Boca Raton, FL, USA: CRC Press, 2011.
[99] B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *J. Statistical Softw.*, vol. 76, no. 1, pp. 1–32, Jan. 2017.
[100] J. Guo, J. Gabry, B. Goodrich, and S. Weber, "RStan: the R interface to stan." [Online]. Available: https://mc-stan.org/rstan/
[101] F. Baker and S. H. Kim, *Item Response Theory: Parameter Estimation Techniques*. Boca Raton, FL, USA: CRC Press, 2004.
[102] D. Ramesh and S. K. Sanampudi, "An automated essay scoring systems: a systematic literature review," *Artif. Intell. Rev.*, Sep. 2021.
[103] S. Watanabe, "Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory," *J. Mach. Learn. Res.*, vol. 11, no. 116, pp. 3571–3594, Dec. 2010.
[104] D.-I. Kim, S. W. Choi, G. Lee, and K. R. Um, "A comparison of the common-item and random-groups equating designs using empirical data," *Int. J. Selection Assessment*, vol. 16, no. 2, pp. 83–92, 2008.
[105] S. Hellman, M. Rosenstein, A. Gorman, W. Murray, L. Becker, A. Baikadi, J. Budden, and P. W. Foltz, "Scaling up writing in the curriculum: Batch mode active learning for automated essay scoring," in *Proc. Sixth ACM Conf. Learning @ scale*, Chicago, IL, USA, Hune, 24–25 2019, pp. 1–10.
[106] Y. Han, W. Wu, Y. Yan, and L. Zhang, "Human-machine hybrid peer grading in SPOCs," *IEEE Access*, vol. 8, pp. 220 922–220 934, 2020.
[107] J. de la Torre, "DINA model and parameter estimation: A didactic," *J. Educ. Behav. Statist.*, vol. 34, no. 1, pp. 115–130, 2009.
[108] F. Wang, Q. Liu, E. Chen, Z. Huang, Y. Yin, S. Wang, and Y. Su, "NeuralCD: A general framework for cognitive diagnosis," *IEEE Trans. Knowl. Data Eng.*, pp. 1–16, 2022.

Masaki Uto received a Ph.D. from the University of Electro-Communications, Tokyo, Japan, in 2013. He has been an associate professor at the University of Electro-Communications since 2020. He received the Best Paper Runner-up Award at AIED2020. His research interests include educational and psychological measurement, Bayesian statistics, machine learning, and natural language processing.

Itsuki Aomi received a master's degree from the University of Electro-Communications, Tokyo, Japan, in 2020. He majored in mathematical sciences and machine learning, and researched probabilistic graphical models and applications of natural language processing. He has been a researcher for Sansan Inc., and now works on R&D for applying machine learning to document data.

Emiko Tsutsumi received a master's degree from the University of Electro-Communications, Tokyo, Japan, in 2019. She is currently pursuing a Ph.D. at the University of Electro-Communications. Her research interests include educational data mining, Bayesian statistics, deep neural networks for adaptive learning, and test theory.

Maomi Ueno received a Ph.D. in Computer Science from the Tokyo Institute of Technology, Tokyo, Japan, in 1994. He has been a Professor of the University of Electro-Communications, Tokyo, Japan, since 2013. His interests are artificial intelligence, machine learning, Bayesian statistics, and Bayesian networks. He is a member of the IEEE.