Neural Automated Essay Scoring Incorporating Handcrafted Features

Masaki Uto
The University of Electro-Communications
uto@ai.lab.uec.ac.jp

Yikuan Xie
The University of Electro-Communications
yikuan.xie@ai.lab.uec.ac.jp

Maomi Ueno
The University of Electro-Communications
ueno@ai.is.uec.ac.jp

Abstract

Automated essay scoring (AES) is the task of automatically assigning scores to essays as an alternative to grading by human raters. Conventional AES typically relies on handcrafted features, whereas recent studies have proposed AES models based on deep neural networks (DNNs) to obviate the need for feature engineering. Furthermore, hybrid methods that integrate handcrafted features in a DNN-AES model have been recently developed and have achieved state-of-the-art accuracy. One of the most popular hybrid methods is formulated as a DNN-AES model with an additional recurrent neural network (RNN) that processes a sequence of handcrafted sentence-level features. However, this method has the following problems: 1) It cannot incorporate effective essay-level features developed in previous AES research. 2) It greatly increases the numbers of model parameters and tuning parameters, increasing the difficulty of model training. 3) It has an additional RNN to process sentence-level features, enabling extension to various DNN-AES models complex. To resolve these problems, we propose a new hybrid method that integrates handcrafted essay-level features into a DNN-AES model. Specifically, our method concatenates handcrafted essay-level features to a distributed essay representation vector, which is obtained from an intermediate layer of a DNN-AES model. Our method is a simple DNN-AES extension, but significantly improves scoring accuracy.

1 Introduction

In various assessment fields, essay-writing tests have attracted much attention as a way to measure practical higher-order abilities such as logical thinking, critical reasoning, and creative-thinking skills (Hussein et al., 2019; Uto, 2019). In essay-writing tests, test-takers are required to write essays about a given topic, and human raters grade those essays based on a scoring rubric. However, because the scoring process takes much time and effort, it is hard to grade large numbers of essays (Hussein et al., 2019). Further, subjectivity in human scoring can reduce accuracy (Amorim et al., 2018; Uto and Ueno, 2018; Uto and Okano, 2020). Automated essay scoring (AES), which utilizes natural language processing and machine learning techniques to automatically grade essays, is one method for resolving these problems.

Many AES methods have been developed over the past decades, and can generally be categorized as feature-engineering and neural-network approaches (Hussein et al., 2019; Ke and Ng, 2019). The feature-engineering approach predicts scores using handcrafted features such as essay length or spelling errors (e.g., (Amorim et al., 2018; Dascalu et al., 2017; Mark D. Shermis, 2016; Nguyen and Litman, 2018)). The advantages of this approach include interpretability and explainability. However, this approach generally requires extensive effort for engineering effective features to achieve high scoring accuracy for various essays.

To obviate the need for feature engineering, a neural-network approach that automatically extracts features using deep neural networks (DNNs) has recently attracted attention. Many DNN-AES models have been proposed and have achieved high accuracy (Alikaniotis et al., 2016; Taghipour and Ng, 2016;

Proceedings of the 28th International Conference on Computational Linguistics, pages 6077–6088
Barcelona, Spain (Online), December 8-13, 2020

Dasgupta et al., 2018; Farag et al., 2018; Jin et al., 2018; Mesgar and Strube, 2018; Wang et al., 2018; Mim et al., 2019; Nadeem et al., 2019; Uto and Okano, 2020).

These two approaches can be viewed as complementary rather than competing because they provide different advantages. Specifically, the neural-network approach can extract dataset-specific features from word sequence patterns, whereas the feature-engineering approach can use existing effective features that are difficult to extract using DNNs from only word sequence information. To obtain both benefits, Dasgupta et al. (2018) proposed a hybrid method that integrates both approaches. This method is formulated as a DNN-AES model with an additional recurrent neural network (RNN) that processes a sequence of handcrafted sentence-level features. This method provides state-of-the-art accuracy, but has the following drawbacks:

1. It cannot incorporate effective essay-level features developed in previous AES research.

2. It greatly increases the numbers of model parameters and tuning parameters, increasing the difficulty of model training.

3. It has an additional RNN that processes sequences of handcrafted sentence-level features, enabling extension to various DNN-AES models complex.

To resolve these problems, we propose a new hybrid method that integrates handcrafted essay-level features into a DNN-AES model. Specifically, our method concatenates handcrafted essay-level features to a distributed essay representation vector, which is obtained from an intermediate layer of a DNN-AES model. The advantages of our method are as follows:

1. It can incorporate various existing essay-level features for which effectiveness has been shown.

2. The number of required additional parameters is only the number of incorporated essay-level features, and there are no additional hand-tuned parameters.

3. It can be easily applied to various DNN-AES models, because conventional models commonly have a layer that produces a distributed essay-representation vector.

Our method is a simple DNN-AES extension, but experimental results on real-world benchmark data show that it significantly improves accuracy.

2 Automated essay scoring methods

This section briefly reviews conventional AES methods based on the feature-engineering and neural-network approaches.

2.1 Feature-engineering approach

Following the first AES method, Project Essay Grade (PEG) (Page, 2003), many feature engineering–based AES methods have been developed, including Intelligent Essay Assessor (IEA) (Foltz et al., 2013), e-rater (Attali and Burstein, 2006), the Bayesian Essay Test Score sYstem (BETSY) (Rudner and Liang, 2002), and IntelliMetric (Schultz, 2013). These methods have been applied to various actual tests. For example, e-rater, a popular commercial AES, now plays the role of a second rater in the Test of English as a Foreign Language (TOEFL) and the Graduate Record Examination (GRE).

These AES methods predict scores by supervised machine learning models using handcrafted features. For instance, PEG and e-rater use multiple regression models, and Phandi et al. (2015) used a correlated Bayesian linear-ridge-regression model. BETSY and Larkey (1998) perform AES using classification models. Other recent works solve AES by using preference-ranking models (Yannakoudakis et al., 2011; Chen and He, 2013).

The features used in previous research differ among the methods, ranging from simple features (e.g., word or essay length) to more complex ones (e.g., readability or grammatical errors). Table 1 shows examples of representative features (Phandi et al., 2015; Ke and Ng, 2019).

Table 1: Representative handcrafted features.

Feature Type	Examples
Length-based features	Numbers of characters, words, sentences, and punctuation symbols. Average word lengths.
Syntactic features	Numbers of nouns, verbs, adverbs, adjectives, and conjunctions. Parse tree depth. Grammatical error rates.
Word-based features	Numbers of useful n-grams and stemmed n-grams. Numbers of spelling errors, sentiment words, and modals.
Readability features	Numbers of difficult words and syllables. Readability indices, such as Flesch–Kincaid reading ease (Kincaid et al., 1975), Gunning fog (Whisner, 2004), or SMOG index (Fitzsimmons et al., 2010).
Semantic feature	Semantic similarity based on latent semantic analysis (Foltz et al., 2013). Histogram-based features computed by pointwise mutual information (Klebanov and Flor, 2013).
Argumentation feature	Numbers of claims and premises. Argument tree depth as estimated using argument mining techniques (Nguyen and Litman, 2018).
Prompt-relevant feature	Number of words in essays for a prompt.

2.2 Neural-network approach

This section introduces two DNN-AES models as AES methods based on the neural-network approach: the most popular model, which uses a long short-term memory (LSTM), and an advanced model based on the transformer architecture.

2.3 LSTM-based model

The LSTM-based model (Alikaniotis et al., 2016), which was the first DNN-AES model, predicts essay scores through the multi-layered neural networks shown in Fig. 1 by inputting essay word sequences. Letting $\mathcal{V} = \{1, \cdots, V\}$ be a vocabulary list, an essay j is defined as a list of vocabulary words $\{\boldsymbol{w}_{ji} \in \mathcal{V} \mid i = \{1, \cdots, n_j\}\}$, where $\boldsymbol{w}_{ji}$ is a V-dimensional one-hot representation of the i-th word in essay j and n_j is the number of words in essay j. This model processes word sequences through the following layers:

Lookup table layer: This layer transforms each word in a given essay into a D-dimensional word-embedding representation, in which words with the same meaning have similar representations. Specifically, letting $\boldsymbol{A}$ be a $D \times V$-dimensional embeddings matrix, the word-embedding representation $\boldsymbol{x}_{ji}$ corresponding to $\boldsymbol{w}_{ji}$ is calculable as the dot product $\boldsymbol{A} \cdot \boldsymbol{w}_{ji}$.

Recurrent layer: This layer is an LSTM network that outputs a vector at each timestep to capture long-distance word dependencies. Specifically, this layer transforms sequence $\{\boldsymbol{x}_{j1}, \boldsymbol{x}_{j2}, \cdots, \boldsymbol{x}_{jn_j}\}$ to an LSTM output sequence $\{\boldsymbol{h}_{j1}, \boldsymbol{h}_{j2}, \cdots, \boldsymbol{h}_{jn_j}\}$. A single-layer unidirectional LSTM is generally used, but bidirectional or multilayered LSTMs are also often used. A convolution neural network is optionally used before the recurrent layer to capture n-gram-level textual dependencies.

Pooling layer: This layer transforms recurrent layer outputs into a fixed-length vector. Mean-over-time (MoT) pooling, which calculates an average vector $\boldsymbol{M}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} \boldsymbol{h}_{ji}$, is generally used. Other frequently used pooling methods include the last pool, which uses the last output of the recurrent layer $\boldsymbol{h}_{jn_j}$, and a pooling-with-attention mechanism.

Linear layer with sigmoid activation: This layer projects pooling-layer output $\boldsymbol{M}_j$ to a scalar value in the range $[0, 1]$ by the sigmoid function $\sigma(\boldsymbol{W}\boldsymbol{M}_j + b)$, where $\boldsymbol{W}$ is a weight matrix and b is a bias.

Model training is conducted by backpropagation with a mean square error (MSE) loss function using a training dataset in which scores are normalized to a $[0, 1]$ scale. During the prediction phase, predicted

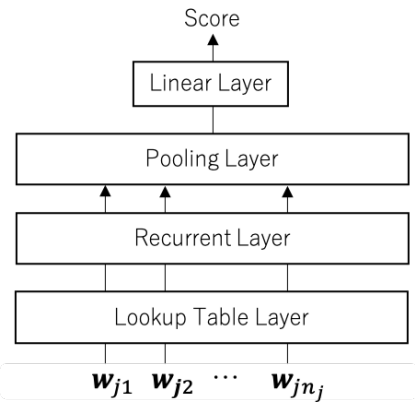

Figure 1: LSTM-based model.

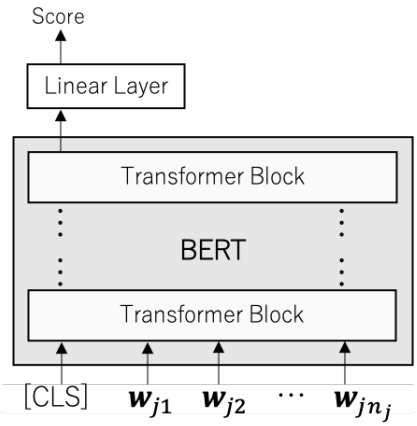

Figure 2: BERT-based model.

scores are rescaled to the original score range. This model has been used as the basis model in various current DNN-AES models (e.g., (Dasgupta et al., 2018; Farag et al., 2018; Jin et al., 2018; Mesgar and Strube, 2018; Wang et al., 2018; Mim et al., 2019; Nadeem et al., 2019; Uto and Okano, 2020)).

2.4 Transformer-based model

Transformer-based architectures have recently attracted attention as an alternative approach to RNN for processing sequential data. Specifically, bidirectional encoder representations from transformers (BERT), a pre-trained multilayer bidirectional transformer network (Vaswani et al., 2017) released by the Google AI Language team, have achieved state-of-the-art results in various NLP tasks, such as question answering, named entity recognition, natural language inference, and text classification (Devlin et al., 2019). BERT was also applied to AES (Rodriguez et al., 2019) and automated short-answer grading (Liu et al., 2019; Sung et al., 2019) in 2019, and demonstrated good performance.

Transformers are a neural network architecture designed to handle ordered data sequences using an attention mechanism. Specifically, transformers consist of multiple layers (called *transformer blocks*), each containing a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. See Ref. (Vaswani et al., 2017) for details of this architecture.

BERT is trained in *pre-training* and *fine-tuning* steps. Pre-training is conducted on huge amounts of unlabeled text data over two tasks, *masked language modeling* and *next-sentence prediction*, the former predicting the identities of words that have been masked out of the input text and the latter predicting whether two given sentences are adjacent.

Using BERT for a target NLP task, including AES, requires fine-tuning (retraining), which is conducted from a task-specific supervised dataset after initializing model parameters to pre-trained values. When using BERT for regression or classification tasks such as AES, input texts require preprocessing, namely, adding a special token ("[CLS]") to the beginning of each text. BERT output corresponding to this token is used as a fixed-length distributed text representation (Devlin et al., 2019). We can thus conduct target regression or classification tasks based on the text representation. In this study, we assume the use of the *linear layer with sigmoid activation*, described in the previous subsection, to predict essay scores from the text representation (Fig. 2).

3 Hybrid method

The feature-engineering approach and the neural-network approach can be viewed as complementary rather than competing approaches, because as mentioned in Section 1 they provide different advantages. To receive both benefits, Dasgupta et al. (2018) proposed a hybrid method that integrates the two approaches.

Figure 3 shows the model architecture of the hybrid method. As that figure shows, it mainly consists of two DNNs. One processes word sequences in a given essay in the same way as the conventional LSTM-

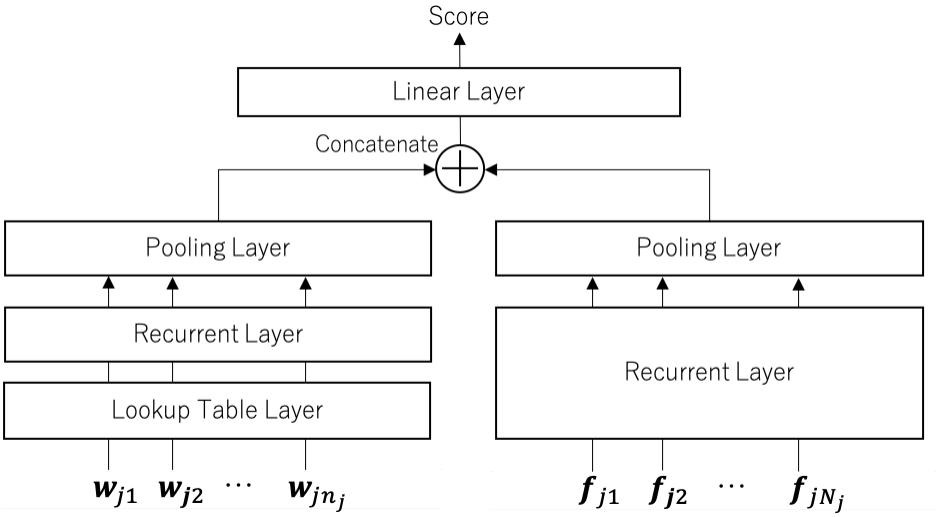

Figure 3: Conventional hybrid model.

based DNN-AES model. Specifically, it transforms a word sequence $\boldsymbol{w}_j = \{\boldsymbol{w}_{j1}, \boldsymbol{w}_{j2}, \cdots, \boldsymbol{w}_{jn_j}\}$ to a hidden vector $\mathcal{H}_j$, which is a fixed-length distributed essay representation, through the lookup table layer, recurrent layer, and pooling layer. The other DNN processes a sequence of handcrafted sentence-level features. Letting the j-th essay have N_j sentences, and letting sentence-level features for the n-th essay sentence be $\boldsymbol{f}_{jn}$, the feature sequence $\boldsymbol{F}_j = \{\boldsymbol{f}_{j1}, \boldsymbol{f}_{j2}, \cdots, \boldsymbol{f}_{jN_j}\}$ is transformed to a fixed-length hidden vector $\mathcal{H}_j^f$ through a recurrent layer and a pooling layer. (Note that the original article used an LSTM for the recurrent layer and attention pooling for the pooling layer.) Finally, inputting a concatenated vector $[\mathcal{H}_j; \mathcal{H}_j^f]$, the linear layer with sigmoid activation produces a predicted score.

This method has provided higher accuracy than feature engineering–based methods or DNN-based methods. However, it has the following drawbacks.

1. It cannot incorporate essay-level features developed in conventional AES research.

2. It has far more model and tuning parameters than does a base DNN-AES model. Specifically, letting the number of handcrafted sentence-level features be f, and the hidden variable size of the LSTM in the recurrent layer be d, this method requires at least $(4df + d^2 + 5d)$ additional parameters, and further parameters are required if attention pooling is used. It also requires tuning parameters for the LSTM and the pooling layer, making model training more difficult.

3. It requires an additional RNN for processing sequences of handcrafted sentence-level features, making implementation with transformer-based models and other DNN-AES models complex.

4 Proposed method

To resolve the above problems, we propose a new hybrid method that incorporates handcrafted essay-level features to a DNN-AES model.

Our method concatenates handcrafted essay-level features to the distributed essay representation $\mathcal{H}_j$, which is the input vector for the last linear layer in conventional DNN-AES models. Letting essay-level features for the j-th essay be $\boldsymbol{F}_j^o$, the proposed method projects the concatenated vector $[\mathcal{H}_j; \boldsymbol{F}_j^o]$ to a scalar value by using a sigmoid function, as in conventional DNN-AES models.

The proposed method can be easily applied to existing DNN-AES models, because they commonly have a layer that produces a distributed essay representation before the last linear layer. As examples, Figs. 4, 5, and 6 show model architectures for LSTM, BERT, and conventional hybrid models integrating essay-level features.

The proposed method can incorporate various existing essay-level features for which effectiveness has been shown. As essay-level features, this study uses the 25 features presented in Table 2, which have

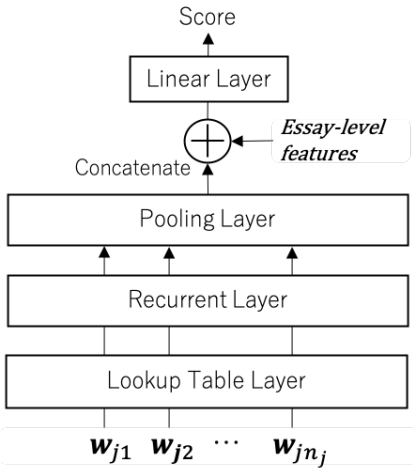

Figure 4: LSTM-based model with essay-level features.

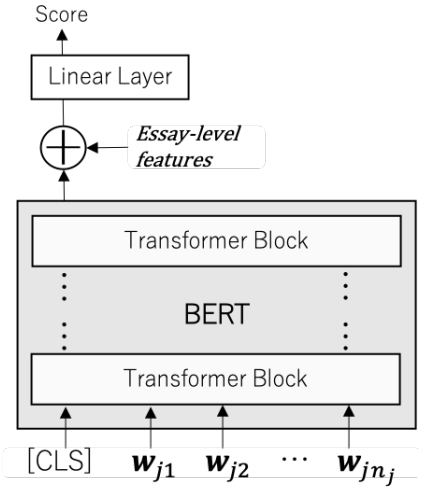

Figure 5: BERT-based model with essay-level features.

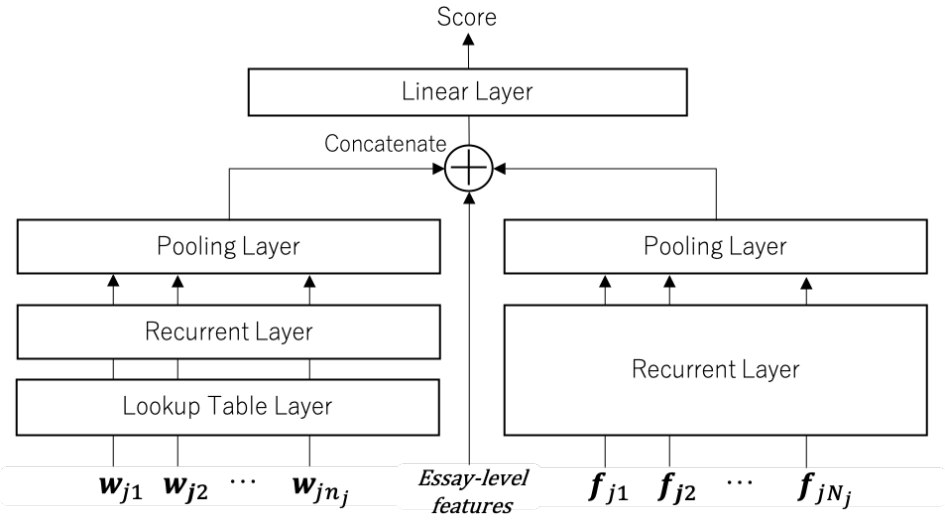

Figure 6: Conventional hybrid model with essay-level features.

been widely used in various AES studies. We assume that the feature values are standardized to fulfill the condition of mean 0 and standard deviation 1.0.

Another advantage of our method is that it requires additional weight parameters in only the last linear layer, and the number of additional parameters is only the number of incorporated essay-level features $\boldsymbol{F}_j^o$, as compared with the basis DNN-AES model. It requires no additional hand-tuned parameters.

5 Experiments

This section demonstrates the effectiveness of the proposed method using real-world benchmark data.

5.1 Experimental procedures

This study employed the automated student assessment prize (ASAP) dataset, which is widely used as benchmark data in AES research. The ASAP dataset provides eight sets of essays, each set associated with a prompt. Essays were written by students in grades 7–10. Table 3 summarizes numbers of essays, score ranges, and averaged essay length for each prompt.

Using this dataset, we evaluated score prediction accuracies through five-fold cross-validation for each prompt. The accuracy metric was the quadratic weighted Kappa (QWK), which examines agreement

Table 2: Essay-level features used in this study.

Feature Type	Features
Length-based features	Numbers of words, sentences, lemmas, and punctuation symbols (commas, exclamation marks, and question marks). Average lengths of words and sentences.
Syntactic features	Numbers of nouns, verbs, adverbs, adjectives, and conjunctions.
Word-based features	Numbers of spelling errors and stop-words.
Readability features	Automated readability index (Smith and Senter, 1967), Coleman–Liau index (Coleman and Liau, 1975), Dale–Chall readability score, difficult word count, Flesch reading ease (Kincaid et al., 1975), Flesch–Kincaid grade (Kincaid et al., 1975), Gunning fog (Whisner, 2004), Linsear write formula, SMOG index (Fitzsimmons et al., 2010), syllable count.

Table 3: Data statistics.

Prompt	# of essays	Score range	Average essay length
1	1783	2–12	350 words
2	1800	1–6	350 words
3	1726	0–3	150 words
4	1770	0–3	150 words
5	1805	0–4	150 words
6	1800	0–4	150 words
7	1568	0–30	250 words
8	721	0–60	650 words

between predicted scores and ground truth. We conducted this experiment for the LSTM-based model (Fig. 1), the BERT-based model (Fig. 2), Dasgupta's hybrid model (Fig. 3), and the proposed method with these models (Figs. 4, 5, and 6). In the LSTM-based model, we used a single-layer LSTM, a two-layer LSTM, and a bidirectional LSTM for the recurrent layer. We used last pooling as the pooling layer for these LSTM-based models, and also examined MoT pooling for the single-layer LSTM-based model. As sentence features for Dasgupta's hybrid model, we used features similar to the essay-level features shown in Table 2 after two modifications: 1) For length-based features, we removed the number and average length of sentences. 2) We removed the SMOG index from the readability features, because it is not definable for a sentence. We also examined a logistic regression model using essay-level features as a method based on the feature-engineering approach.

We implemented the models in the Python programming language with the Keras library. As the embedding matrix, we used Glove (Pennington et al., 2014) with 50 dimensions. We set LSTMs' hidden-variable dimension to 300, the mini-batch size to 32, and the maximum epochs to 50. We used dropout regularization to avoid overfitting, with dropout probabilities for lookup table layer output and pooling layer output set to 0.5. The recurrent dropout probability was set to 0.2. We used the Adam optimization algorithm (Kingma and Ba, 2014) to minimize the mean squared error (MSE) loss function over the training data. For the BERT model, we used a *base*-sized pre-trained model.

5.2 Experimental results

Table 4 shows the experimental results.

Comparing accuracy among prompts, accuracy tends to be higher for prompts in which the average essay length is short than those with long essays. For example, the accuracy for prompts 4, 5, 6, and 7 tends to be higher than that for prompts 2 and 8 in each model. This tendency is consistent with previous

Table 4: Experimental results.

	Prompt									
	1	2	3	4	5	6	7	8	Avg.	p-value
LSTM	0.373	0.407	0.516	0.773	0.753	0.767	0.635	0.174	0.550	0.018
+ Essay-level features	0.801	0.621	0.602	0.778	0.771	0.777	0.761	0.645	0.720	
LSTM with MoT	0.717	0.522	0.616	0.775	0.796	0.783	0.749	0.562	0.690	0.015
+ Essay-level features	0.821	0.649	0.617	0.790	0.787	0.807	0.794	0.694	0.745	
2-layer LSTM	0.435	0.414	0.530	0.791	0.698	0.768	0.639	0.163	0.555	0.017
+ Essay-level features	0.778	0.620	0.592	0.779	0.779	0.769	0.762	0.643	0.715	
Bidirectional LSTM	0.484	0.419	0.500	0.777	0.738	0.721	0.625	0.218	0.560	0.014
+ Essay-level features	0.779	0.597	0.582	0.778	0.762	0.765	0.756	0.661	0.710	
BERT	0.829	0.391	0.762	0.886	0.876	0.584	0.818	0.540	0.711	0.021
+ Essay-level features	**0.852**	0.651	**0.804**	**0.888**	**0.885**	**0.817**	**0.864**	0.645	**0.801**	
Conventional hybrid	0.729	0.635	0.631	0.787	0.802	0.793	0.773	0.693	0.730	0.073
+ Essay-level features	0.823	**0.674**	0.601	0.795	0.790	0.811	0.806	**0.714**	0.752	
Logistic regression	0.822	0.648	0.666	0.704	0.783	0.672	0.724	0.600	0.702	-

studies.

Comparing the conventional DNN-AES models shows that the LSTM-based model with MoT pooling has higher performance than models with last pooling, which is also consistent with previous studies (Alikaniotis et al., 2016; Riordan et al., 2017). BERT tends to outperform the LSTM-based models, as in other BERT applications including automated short-answer grading (Devlin et al., 2019; Liu et al., 2019; Lun et al., 2020; Sung et al., 2019). As Dasgupta et al. (2018) reported, the conventional hybrid model shows the highest average accuracy among the conventional models.

Table 4 shows that by incorporating handcrafted essay-level features, the proposed method drastically improves accuracy of all base DNN-AES models. We conducted paired t-tests to examine whether averaged performance of the proposed method is significantly higher than base model performance. The results, shown in the "p-value" column in Table 4, indicate that the proposed method improved performance at the 5% significance level for the LSTM- and BERT-based models, and at the 10% significance level for the conventional hybrid model.

Comparing the proposed methods with the logistic regression model (a feature-engineering approach), all of the proposed methods provided a higher average accuracy. The paired t-test between the logistic regression model and the proposed method shows that averaged QWKs of the proposed method using LSTM with MoT pooling and the conventional hybrid model were higher at the 5% significance level, and that of the BERT-based proposed method was higher at the 1% significance level.

Among the proposed methods, the one using the BERT model provided the highest average accuracy.

To confirm whether the handcrafted essay-level features were effective, Table 5 shows weight parameter values in the final linear layer of the BERT-based proposed model. In the table, the row *Distributed representation* shows the average values of the absolute weight parameters for the 300-dimensional essay distributed representation vector $\mathcal{H}_j$. A higher weight value means that the feature has more influence on score prediction. This table suggests that each handcrafted feature contributes to some extent, whereas features with large weights vary across prompts.

These experimental results show that the proposed method effectively improves AES accuracy.

6 Conclusions

We proposed a simple method that incorporates handcrafted essay-level features to DNN-AES models. Our method adds handcrafted features to a distributed essay representation vector obtained as an intermediate hidden representation of a DNN-AES model. Our method can be easily applied to various conventional DNN-AES models without increasing model complexity much, but significantly improving

Table 5: Feature weights for the BERT-based proposed model.

	Prompt							
	1	2	3	4	5	6	7	8
Length-based features								
# of words	0.018	-0.087	0.393	0.123	-0.117	-0.296	0.366	-0.196
# of sentences	-0.123	0.151	0.078	0.033	0.209	0.130	0.335	0.050
# of lemmas	0.073	0.026	0.168	-0.149	0.159	0.406	0.387	0.219
# of commas	0.055	0.048	0.060	-0.022	0.030	0.002	0.043	0.041
# of exclamation marks	0.021	-0.005	-0.046	-0.108	0.003	-0.020	0.003	-0.019
# of question marks	0.062	0.012	-0.040	-0.026	0.003	0.008	-0.061	-0.034
Avg. word length	0.351	0.013	0.081	-0.253	0.234	0.163	-0.353	0.060
Avg. sentence length	0.076	0.017	-0.106	-0.152	-0.012	0.033	0.007	-0.035
Syntactic features								
# of nouns	0.226	-0.002	0.012	0.321	0.280	0.285	-0.009	-0.089
# of verbs	0.140	0.111	0.041	-0.003	0.098	0.079	-0.061	0.115
# of adjectives	0.031	-0.010	-0.037	0.271	-0.011	0.344	0.000	0.046
# of adverbs	0.060	0.035	-0.032	-0.084	0.020	0.140	-0.020	0.045
# of conjunctions	0.012	-0.027	0.138	-0.002	0.047	-0.133	0.000	0.057
Word-based features								
# of spelling errors	0.001	-0.058	-0.077	0.014	0.038	-0.165	-0.085	-0.043
# of stop-words	-0.113	0.039	-0.147	-0.062	0.446	0.291	-0.126	-0.335
Readability features								
Automated readability index	0.019	0.238	0.286	0.307	0.147	-0.100	-0.005	-0.038
Coleman–Liau index	-0.366	0.049	-0.159	0.144	-0.053	-0.072	0.293	-0.134
Dale–Chall readability score	0.009	-0.207	0.043	0.096	-0.002	-0.031	0.044	0.003
Difficult word count	0.139	0.202	0.315	0.279	-0.171	0.140	-0.005	0.076
Flesch reading ease	0.078	-0.166	-0.042	0.219	-0.058	-0.219	-0.050	-0.035
Flesch–Kincaid grade	-0.002	0.134	-0.076	-0.019	-0.182	0.135	-0.030	0.082
Gunning fog	-0.075	-0.301	0.002	-0.210	0.296	-0.195	-0.010	-0.038
Linsear write formula	0.032	-0.067	-0.151	0.195	-0.163	-0.007	-0.054	-0.021
Smog index	0.090	0.063	-0.046	0.203	0.054	0.081	0.106	0.071
Syllables counts	0.166	0.048	0.261	0.506	-0.055	-0.339	-0.352	0.289
Distributed representation†	0.046	0.043	0.050	0.050	0.044	0.049	0.036	0.039

†: Averaged absolute weights for 300-dimensional essay distributed representation

prediction performance.

In this study, we evaluated the effectiveness of the proposed method that uses relatively simple features, but in future studies, we will use more varied essay-level features, such as those shown in Table 1. Additionally, we will conduct an ablation experiment on essay-level features to clarify which features are effective for which DNN-AES models. Another future aim is to apply the proposed method to more varied DNN-AES models, such as those mentioned in Subsection 2.3. Moreover, although our method directly adds essay-level features to the DNN-based distributed essay representation vector, accuracy might be further improved by appending several layers after the feature input layer. Such model extensions are also another topic for future study.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers JP17H04726, JP19H05663, JP19K21751, and JP20K20817.

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 715–725.

Evelin Amorim, Márcia Cançado, and Adriano Veloso. 2018. Automated essay scoring in the presence of biased ratings. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 229–237.

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater v.2. *The Journal of Technology, Learning and Assessment*, 4(3):1–31.

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Mihai Dascalu, Wim Westera, Stefan Ruseti, Stefan Trausan-Matu, and Hub Kurvers. 2017. Readerbench learns dutch: Building a comprehensive automated essay scoring system for Dutch language. In *Proceedings of the International Conference on Artificial Intelligence in Education*, pages 52–63.

Tirthankar Dasgupta, Abir Naskar, Lipika Dey, and Rupsa Saha. 2018. Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In *Proceedings of the Workshop on Natural Language Processing Techniques for Educational Applications, Association for Computational Linguistics*, pages 93–102.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. 2018. Neural automated essay scoring and coherence modeling for adversarially crafted input. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 263–271.

Paul R Fitzsimmons, BD Michael, Joane L Hulley, and G Orville Scott. 2010. A readability assessment of online parkinson's disease information. *The journal of the Royal College of Physicians of Edinburgh*, 40(4):292–296.

Peter W. Foltz, Lynn A. Streeter, and Karen E. Lochbaum. 2013. Handbook of automated essay evaluation: Current applications and new directions. In *Implementation and Applications of the Intelligent Essay Assessor*. Routledge.

Mohamed Abdellatif Hussein, Hesham A. Hassan, and Mohamed Nassef. 2019. Automated language essay scoring systems: A literature review. *PeerJ Computer Science*, 5:e208.

Cancan Jin, Ben He, Kai Hui, and Le Sun. 2018. TDNN: A two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1088–1097.

Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 6300–6308.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Institute for Simulation and Training, University of Central Florida.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Beata Beigman Klebanov and Michael Flor. 2013. Word association profiles and their use for automated scoring of essays. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1148–1158.

Leah S. Larkey. 1998. Automatic essay grading using text categorization techniques. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 90–95.

Tiaoqiao Liu, Wenbiao Ding, Zhiwei Wang, Jiliang Tang, Gale Yan Huang, and Zitao Liu. 2019. Automatic short answer grading via multiway attention networks. In *Proceedings of the International Conference on Artificial Intelligence in Education*, pages 169–173.

Jiaqi Lun, Jia Zhu, Yong Tang, and Min Yang. 2020. Multiple data augmentation strategies for improving performance on automatic short answer scoring. In *Proceedings of the Association for the Advancement of Artificial Intelligence*.

Jill C. Burstein Mark D. Shermis. 2016. *Automated Essay Scoring: A Cross-disciplinary Perspective*. Taylor & Francis.

Mohsen Mesgar and Michael Strube. 2018. A neural local coherence model for text quality assessment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 4328–4339.

Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi, and Kentaro Inui. 2019. Unsupervised learning of discourse-aware text representation for essay scoring. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 378–385.

Farah Nadeem, Huy Nguyen, Yang Liu, and Mari Ostendorf. 2019. Automated essay scoring with discourse-aware neural models. In *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics*, pages 484–493.

Huy V. Nguyen and Diane J. Litman. 2018. Argument mining for improving the automated scoring of persuasive essays. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, pages 5892–5899.

E. B. Page. 2003. Project essay grade: PEG. In *Automated essay scoring: A cross disciplinary perspective*. Lawrence Erlbaum Associates.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1532–1543.

Peter Phandi, Kian Ming A. Chai, and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 431–439.

Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chong Min Lee. 2017. Investigating neural architectures for short answer scoring. In *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics*, pages 159–168.

Pedro Uria Rodriguez, Amir Jafari, and Christopher M. Ormerod. 2019. Language models and automated essay scoring. arXiv, cs.CL.

Lawrence Rudner and Tahung Liang. 2002. Automated essay scoring using bayes' theorem. *Journal of Technology, Learning, and Assessment*, 1, 08.

Matthew. T Schultz. 2013. The intellimetric automated essay scoring engine: A review and an application to chinese essay scoring. In *Handbook of automated essay evaluation: Current applications and new directions*. Routledge.

Edgar A Smith and R. J. Senter. 1967. Automated readability index. Technical report, Cincinnati University, OH.

Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. 2019. Improving short answer grading using transformer-based pre-training. In *Proceedings of the International Conference on Artificial Intelligence in Education*, pages 469–481.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891.

Masaki Uto and Masashi Okano. 2020. Robust neural automated essay scoring using item response theory. In *Proceedings of the International Conference on Artificial Intelligence in Education*, pages 549–561.

Masaki Uto and Maomi Ueno. 2018. Empirical comparison of item response theory models with rater's parameters. *Heliyon, Elsevier*, 4(5):1–32.

Masaki Uto. 2019. Rater-effect IRT model integrating supervised LDA for accurate measurement of essay writing ability. In *Proceedings of the International Conference on Artificial Intelligence in Education*, pages 494–506.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*, pages 5998–6008.

Yucheng Wang, Zhongyu Wei, Yaqian Zhou, and Xuanjing Huang. 2018. Automatic essay scoring incorporating rating schema via reinforcement learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 791–797.

Mary Whisner. 2004. When judges scold lawyers. *Law Libr. J.*, 96:557.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189.