Book Chapter

Improving Performance of Automated Essay Scoring by Using Back-Translation Essays and Adjusted Scores

You-Jin Jong[1], Yong-Jin Kim[2]*, Ok-Chol Ri[1] and Kum-Sok Sin[3]

[1]Kum Sung Middle School Number 2, Pyongyang 999093, Democratic People Republic of Korea
[2]Faculty of Mathematics, KIM IL SUNG University, Pyongyang 999093, Democratic People Republic of Korea
[3]PSJDC Institute, Democratic People Republic of Korea

***Corresponding Author:** Yong-Jin Kim, Faculty of Mathematics, KIM IL SUNG University, Pyongyang 999093, Democratic People Republic of Korea

Published **October 06, 2022**

This Book Chapter is a republication of an article published by Yong-Jin Kim, et al. at Mathematical Problems in Engineering in June 2022. (You-Jin Jong, Yong-Jin Kim, Ok-Chol Ri. Improving Performance of Automated Essay Scoring by Using Back-Translation Essays and Adjusted Scores. Mathematical Problems in Engineering. Volume 2022, Article ID 6906587, 10 pages. https://doi.org/10.1155/2022/6906587)

How to cite this book chapter: You-Jin Jong, Yong-Jin Kim, Ok-Chol Ri, Kum-Sok Sin. Improving Performance of Automated Essay Scoring Using Back-Translation and Score Adjustment. In: Ibrahim Nonkane, editor. Prime Archives in Applied Mathematics: 3rd Edition. Hyderabad, India: Vide Leaf. 2022.

Data Availability: The ASAP dataset used in this study is available at https://www.kaggle.com/c/asap-aes. The Python codes and back-translation data used to support the findings of this study are available at https://github.com/j-y-j-109/asap-back-translation and https://github.com/sdeva14/sustai21-counter-neural-essay-length.

Disclosure: A preprint of this paper can be found at https://arxiv.org/abs/2203.00354 [30].

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

Abstract

Automated essay scoring plays an important role in judging students' language abilities in education. Traditional approaches use handcrafted features to score and are time-consuming and complicated. Recently, neural network approaches have improved performance without any feature engineering. Unlike other natural language processing tasks, only a small number of datasets are publicly available for automated essay scoring, and the size of the dataset is not sufficiently large. Considering that the performance of a neural network is closely related to the size of the dataset, the lack of data limits the performance improvement of the automated essay scoring model. In this paper, we proposed a method to increase the number of essay-score pairs using back-translation and score adjustment and applied it to the Automated Student Assessment Prize dataset for augmentation. We evaluated the effectiveness of the augmented data using models from prior work. In addition, performance was evaluated in a model using long short-term memory, which is widely used for automated essay scoring. The performance was improved by using augmented data.

Introduction

Artificial intelligence is one of the key drivers of industrial development and is an important factor in accelerating the integration of emerging technologies. Nowadays, artificial

intelligence is being used in a very wide way in almost all aspects of our lives [1-4].

Currently, online education systems are being used more actively due to the COVID 19 outbreak, and the role of educational assessment systems has become more important. Learning a foreign language has become more common. Learning a foreign language is not only to satisfy our interests and flirts. In today's multicultural society, it has become essential to freely speak a second or third language. Writing is an important part of language learning. The assessment of writing ability is included in all language tests.

Automated Essay Scoring (AES) is the task of evaluating one's writing ability and assigning a score to an essay without human interference. The process of manually scoring essays is complex and time-consuming. Even though there is a fixed scoring guide, the scoring process is influenced by individual factors, such as mood and personality, and assigned scores are subjective and lack credibility. Automatic scoring for the essay was proposed as a solution to manual scoring.

All early works were based on feature engineering, and the performance was improved by adding more complicated features. In recent years, neural network models have been introduced in this research and have improved the performance without any feature engineering. Various neural networks have been used for AES, and their performance has been continuously improved. From the simplest Recurrent Neural Network (RNN) [5] and Convolutional Neural Network (CNN) to a complicated large-scale natural language pretraining model (XLNet) [6], almost every neural network has been used for AES. Prior works have improved performance by changing the structure of the neural network model or adding other features.

However, we improved the performance by generating more useful data from the original data. Data augmentation techniques have been applied to other natural language processing (NLP) tasks and have shown good performance. However, there are no

examples of data augmentation techniques applied to AES. Our study is the first attempt to augment AES data.

Data augmentation for the NLP corpus should be conducted at the sentence level or the document level. If we just replaced some words, some information in the entire text may be lost. Therefore, back translation, which is simple and augments data at the document level, was selected as our data augmentation technique.

We generated back-translation essays using Google Translator (https://translate.google.com/) and adjusted the corresponding scores in several ways. We trained and validated the model with doubled number of essay-score pairs and tested it on the original data. The performance is improved by using augmented data.

Our main contributions are as follows.

1. Data augmentation was introduced into AES and improved performance. We proved the possibility of data augmentation by adjusting the score along with the essay.
2. We analyzed the characteristics of back-translation essays and the AES task and came up with a score adjustment method suitable for back-translation essays in AES data.
3. We generated back-translation essays (English-Chinese-English, English-French- English) (https://github.com/j-y-j-109/asap-back-translation) for the Automated Student Assessment Prize (ASAP) dataset (https://www.kaggle.com/c/asap-aes).

Related Work

The first AES system was created in 1966 and uses some linguistic features to score essays [7]. Most recent works have used neural network models for AES. In 2016, Taghipour and Ng [8] designed a neural network model using CNN and Long Short-Term Memory (LSTM) [15] and showed significant improvement compared to traditional methods that depend on manual feature engineering (see Figure 1). This is the simplest and most representative model, and it generates a representation

of the input essay and obtains a value from it. The convolution layer extracts local features from the essay and the recurrent layer generates a representation for an essay. In the mean over time layer, the sum of outputs of the recurrent layer is divided into essay length. Let $\boldsymbol{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_N)$ be the outputs of the recurrent layer. Then the function of the Mean over time layer is defined by Equation 1:

$$MoT(\boldsymbol{H}) = \frac{1}{N}\sum_{t=1}^{N} \boldsymbol{h}_t \quad (1)$$

In the last linear layer, they used the sigmoid function to get a score in the range of (0,1). The sigmoid function is given by Equation 2:

$$s(\boldsymbol{H}) = sigmoid(\boldsymbol{w} \cdot MoT(\boldsymbol{H}) + b) \quad (2)$$

Therefore, they normalized all scores to [0,1] before training the model.

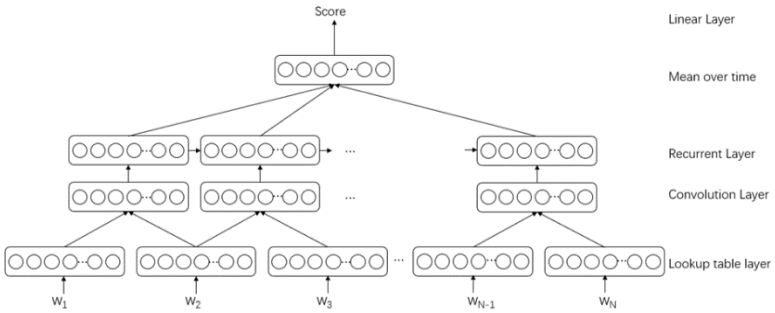

Figure 1: Architecture of the model in [8].

Since then, an increasing number of neural network models have been used for AES, and almost all neural networks including GRU [16], BERT [17], and XLNet have been used (see Table 1). RNN was also used, but it was not used in the final model because the performance was lower than GRU or LSTM.

After simple models, such as the model in [8], models that assign scores by capturing some other features, such as coherence and relevance, have also been used. Example essays were also used

to assign a score to the input essay. In [12] and [14], the relevance between the prompt and the essay and the coherence of sentences within the essay were captured. In [11], the similarity between the word distribution of the input essay and that of the example essays was used. In [12], the k-means algorithm was used to classify example essays, and representative essays were selected from each cluster and used to assign scores.

Various models and embeddings have been used as representations for words and sentences. In [8], the pre-trained word embeddings released in [18] were used. In [9], C&W embeddings were used [19,20] and augmented by considering the contribution of each word to the essay score. In [10], character embeddings were attempted, and in [10] and [11], they used pre-trained Glove embeddings trained on Google News [21]. In [12] and [14], BERT was used to obtain sentence representations.

Data augmentation is a technique to increase the size of data by slightly modifying the original data or adding newly created data to the original data. Data augmentation has received a lot of attention as it reduces the number of cases where neural network training fails due to a lack of data. It can also improve the performance of neural network models. For example, data augmentation such as flipping and rotating an image is used in the field of computer vision. Unlike images, natural language is discrete and it is more difficult to augment data, but many data augmentation techniques have been proposed recently.

Table 1: Various neural network models used for AES.

	CNN	GRU	LSTM	Bi-LSTM	BERT	XLNet
[8]	√		√			
[9]				√		
[10]	√		√			
[11]		√				√
[12]			√			
[13]					√	
[14]			√			

Data augmentation techniques are widely used in NLP tasks. For the Text Classification task, Zhang et al. [22] used the method of finding synonyms in WordNet and replacing words, and for the Categorization task, Wang et al. [23] obtained synonyms by calculating cosine similarity. Yu et al. [24] generated augmented data by translating sentences into French and again into English for Reading Comprehension, and Lun et al. [25] generated back-translation data using Japanese for Automatic Short Answer Scoring. To date, various data augmentation techniques have been proposed and used for many NLP tasks. However, no prior works have applied data augmentation to AES.

Back-translation means that the original data are translated into other languages and then translated back to obtain new data in the original language. This method rewrites the entire text without replacing individual words. In [24] and [26], the English-French translation model was used to perform back-translation for each sentence. In addition to the trained machine translation model, Google's Cloud Translation API service has been widely used [27,28]. In [25], the Baidu Translation API service was used. There are also other methods to add various additional features based on back-translation.

Augmented Data

This section describes the original data and augmented data in detail. Dataset for AES consists of essays and corresponding scores. Therefore, when creating new data using data augmentation techniques, essays and corresponding scores should be determined together.

Original Dataset

There are several open datasets for AES, and more than 90% of prior works were evaluated using the ASAP dataset [29]. In 2012, Kaggle hosted the ASAP competition to evaluate the capabilities of AES systems. The ASAP dataset is built with essays written by students ranging in grade levels from Grade 7 to Grade 10. There are approximately 13,000 essays corresponding to 8 prompts. For individual prompts, the number

of essays is less than 2,000. Specific dataset information is presented in Table 2. Each prompt has a different score range and number of essays. The test set used in the competition is not publicly available.

Table 2: Statistics of ASAP dataset.

Prompt	Number of Essays	Score Range
1	1783	2-12
2	1800	1-6
3	1726	0-3
4	1772	0-3
5	1805	0-4
6	1800	0-4
7	1569	0-30
8	723	0-60

Identifying information from the essays of the ASAP dataset was removed using the Named Entity Recognizer from the Stanford Natural Language Processing group and a variety of other approaches. The relevant entities were identified in the essay and then replaced with a string starting with '@'. Any misspelled words or grammatical errors were transcribed exactly as they occur in the original essays.

Back-Translation

First, we need to obtain back-translation essays using essays of the original data. To study the general effect of back-translation essays, we generated back-translation essays using two languages. We hypothesized that back-translation using different languages would help to diversify augmented data. We used multibyte language, Chinese, and single-byte language, French.

As we mentioned in the Related work section, in [27] and [28], the data was augmented by using the Google Translate API. However, we did not use the Google Translate API, but we used the homepage version of Google Translator. You don't need to write any additional code, and you don't have to make as many requests as you do when using the API. By dividing the entire

text into documents of a size that Google Translator can process at once, the translation can be completed by requesting the number of documents times.

The reasons we used Google Translator are as follows. First, the quality of the back-translation is related to the translator. Our study below assumes that the quality of the translator is good enough. Second, our results must be reproducible. Google Translator is a popular translator that anyone can use.

As the amount of data that Google Translator can process at one time is limited, the original essays were divided into 8 equal-sized parts for translation. Therefore, the entire data was completed by requesting 32(8 (number of parts) * 2 (back-translation) * 2 (two languages)) times of translation. Google Translator perfectly translated special words starting with @ in essays.

Score Adjustment

After obtaining the back-translation essays, the corresponding scores must be determined. The score setting directly affects the performance of data augmentation. This is because even if the number of essays in the data is large, the performance of the model can be further degraded if the scores for essays are not reasonably determined (see Evaluation Section). Therefore, it is essential to adjust the scores for back-translation essays.

The most intuitive way to set the score is to give the score of the original essay since back-translation essays are similar to original essays. In this case, the new scores are given by Equation 3:

$$s_{new}^{d} = s_{ori}^{d}, d \in E_i, i = \overline{1..8} \quad (3)$$

where E_i represents all essays of prompt i, s_{ori}^{d} and s_{new}^{d} are the original and new score of essay d respectively. Another method provides a more suitable score by finely adjusting the original score. In this case, the new scores are given by Equation 4 or 5:

$$\underset{1..8}{s_{new}^{d}} = \min\left(s_{max}^{i}, s_{ori}^{d} + \mathrm{v}\right), d \in \{e|P(e) = 1 \cap e \in E_i\}, i = \quad (4)$$

or

$$\underset{1..8}{s_{new}^{d}} = \max\left(s_{min}^{i}, s_{ori}^{d} - \mathrm{v}\right), d \in \{e|P(e) = 1 \cap e \in E_i\}, i = \quad (5)$$

where P is a condition. For example, let P be the condition for judging whether an essay has a length greater than 300. $P(e) = 1$ means that the length of essay e is greater than 300. s_{max}^{i} and s_{min}^{i} means the maximum and minimum scores that an essay in prompt i can take. v is an additional value to adjust the score.

The essays in the ASAP dataset have certain characteristics. In the essays of the ASAP dataset, certain errors, such as grammatical and lexical errors, exist because of the characteristics of the AES task. For example, as described in section 3.1, there are some misspelled words in the ASAP dataset, and the number of misspelled words decreases after back-translation using Chinese (see Figure 2). If you use a translator, the translator can correct these errors to a certain extent in the process of translating them into other languages, and you can generate translated essays with a smaller number of errors (see Figure 3). Here we assumed that the translator is good enough to do so. When generating back-translation essays using these translated essays, the translator can generate essays at a relatively higher level than the original essays. Therefore, it can be assumed that the quality of back-translation essays is slightly higher than that of original essays.

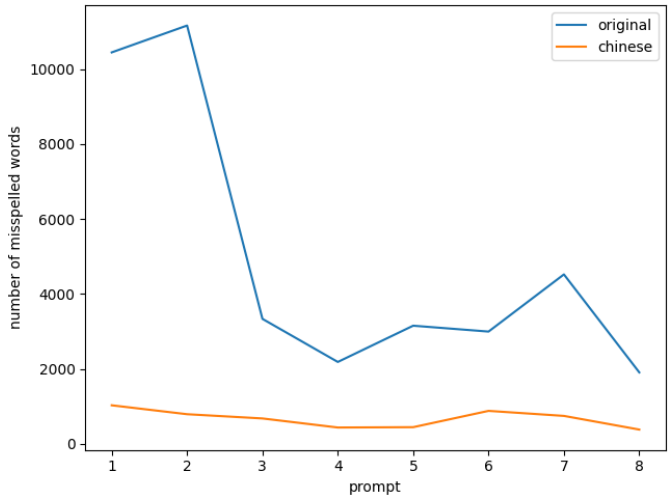

Figure 2: Number of words (the words are separated by word tokenizer of the Natural Language Tool Kit) undefined in Glove for each prompt.

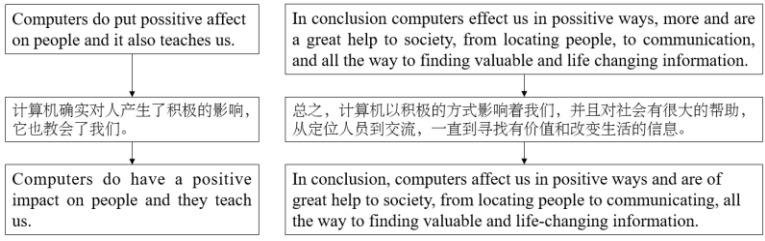

Figure 3: Results of back-translation for sentences from the ASAP dataset (essay_id: 114 and 141).

Each prompt in the ASAP dataset has a different score range (see Figure 4 and Table 3).

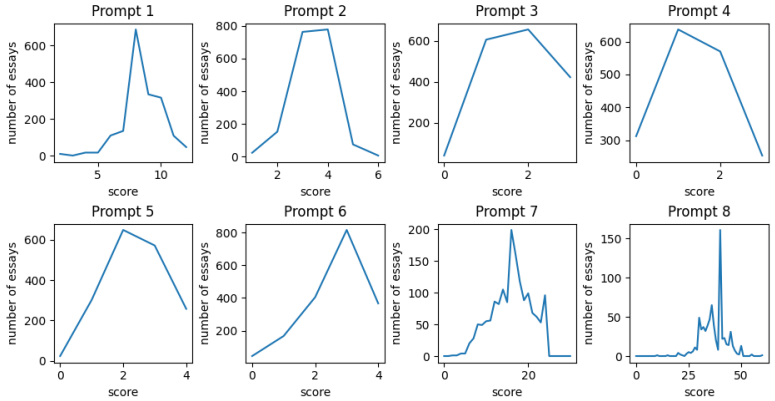

Figure 4: Score distribution for each prompt.

Table 3: Highest frequency score and number of lower/higher score essays for each prompt.

Prompt	Score Range	Highest frequency Score	Number of lower score essays	Number of higher score essays
1	2-12	8	977	806
2	1-6	4	1718	82
3	0-3	2	1303	423
4	0-3	1	949	823
5	0-4	2	975	830
6	0-4	3	1433	367
7	0-30	16	825	744
8	0-40	40	577	146

Considering that back-translation essays are similar to the original essays, scores of back-translation essays were given as original scores according to Equation 3 for all prompts. For prompts with a small score range, that is, 1-6, back-translation cannot change the score of essays. For example, if there are only two scores, 0 and 1, an essay with a score of 0 cannot be back-translated to a score of 1, and an essay with a score of 1 cannot be back-translated to a score of 0. A change of 1 point requires large changes in the essays. For prompts 7 and 8, Equation 4 was also used to determine the scores of back-translation essays. Condition P returns 1 when the essay has a score higher than the highest frequency score, and v is given as 1. For prompts 7 and 8, the scores of back-translation essays are given by Equation 6:

$$s_{new}^{d} = \min\left(30, s_{ori}^{d} + 1\right), d \in \{e | s_{ori}^{e} > 16 \cap e \in E_7\} \quad (6)$$

$$s_{new}^{d} = \min\left(60, s_{ori}^{d} + 1\right), d \in \{e | s_{ori}^{e} > 40 \cap e \in E_8\}$$

First, the additional point is set to 1 because back-translation slightly raises the quality of the essay. Second, when scoring, a higher score is usually given for a more perfectly written essay than the baseline. If you want to obtain a high score, you have to complete it more precisely in all aspects, such as vocabulary and grammar. The baseline can be assumed as the level of the essay with the highest frequency score. For high-score essays, even if there is a slight improvement, the score increases. In other words, the scores of back-translation essays from low-score essays do not increase even if back-translation is performed, but the scores of back-translation essays from high-score essays do increase after back-translation. For each prompt, the highest frequency score was found, and additional points were given to essays with a score higher than that score.

Evaluation

We hypothesized that we could show the generality of back-translation by using French and Chinese. Also, on the assumption that the quality of the translator is good, the score adjustment method was determined. In this section, we evaluate whether the performance of the model improves and whether our score adjustment method is effective.

The models proposed in [11] were used. This model is a recent model that is very similar to the model in [8], and XLNet is also used as a recurrent layer.

To evaluate the effectiveness of the augmented data and for a fair comparison, we trained the models using their published code (https://github.com/sdeva14/sustai21-counter-neural-essay-length). The effectiveness of data augmentation was evaluated by training the model using the original data and augmented data. Performance was evaluated with Quadratic Weighted Kappa (QWK). The QWK was the official criterion for the

ASAP competition and was used to evaluate and compare the performance of models in many works. W is a matrix constructed by Equation 7:

$$\boldsymbol{W}_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{7}$$

where i and j are the gold score and predicted score respectively, N is the number of possible scores. The QWK score is calucated by using Equation 8:

$$QWK = 1 - \frac{\sum_{i,j} \boldsymbol{W}_{i,j} \boldsymbol{O}_{i,j}}{\sum_{i,j} \boldsymbol{W}_{i,j} \boldsymbol{E}_{i,j}} \tag{8}$$

Here $O_{i,j}$ represents the number of essays received score j by model and score i by human. The matrix E is the outer product of histogram vectors of the two scores. To make the sum of elements in O and E the same, matrix E is normalized.

Model

We used two models to determine whether the augmented data improves the model's performance (see Figure 5). As the first model, we use 'Manipulating-Length-GRU'. This model does not divide the sum of outputs of the recurrent layer by the length of the essay, as in the model in [8] or other models, but by the average length of the essays included in each prompt. GRU was used as the recurrent layer. The function of the layer is given by Equation 9:

$$Avg_1^i(\boldsymbol{H}) = \frac{1}{N_{avg}^i} \sum_{t=1}^{N} \boldsymbol{h}_t, i = \overline{1..8} \tag{9}$$

$N_{avg}$is the average length of the essays included in prompt i.

As the second model, 'Considering-Content-LSTM' was used. This model computes the KL divergence between the word distribution of the example essays divided into three levels and the word distribution of the input essay and concatenates them to the averaged output of the recurrent layer. In this model, the function of the layer is given by Equation 10:

$$Avg_2^i(\boldsymbol{H}) = [Avg_1^i(\boldsymbol{H}); KL_1 KL_2 KL_3], i = \overline{1..8} \tag{10}$$

KL_1, KL_2 and KL_3 are KL divergences for three levels respectively.

In [11], GRU and XLNet were used, but we used LSTM, which is widely used for AES.

Since the complexity of the recurrent layer is the biggest in the model, the complexity of the recurrent layer becomes the complexity of the model. Therefore, the complexity of the model is $O(D^2N)$. D is the output dimension of the recurrent layer and N is the length of the input essay.

As a word vector, these models used Glove, a 100-dimensional pre-trained embedding model trained on Google News.

Experimental Setup

The ASAP dataset does not have a test set, and cross-validation is used to evaluate the models. We used the same cross-validation partitions as those in [8]. We trained and validated the model with doubled number of essay-score pairs and evaluated performance on the original test set. We performed 50 epochs on the validation set and applied the best model to the test set. ADAM optimizer (eps=1e-7) was used with a learning rate of 0.001. The batch size was set to 32. The cell size of the recurrent layer was 300. The loss function is given by Equation 11:

$$MSE(s, s^*) = \frac{1}{N_A}\sum_{i=1}^{N_A}(s_i - s_i^*)^2 \quad (11)$$

N_A is the number of essays in dataset. s_i is a gold score and s_i^* is a predicted score.

We have performed 15 times of training for every prompt and obtained the performance value as an average value.

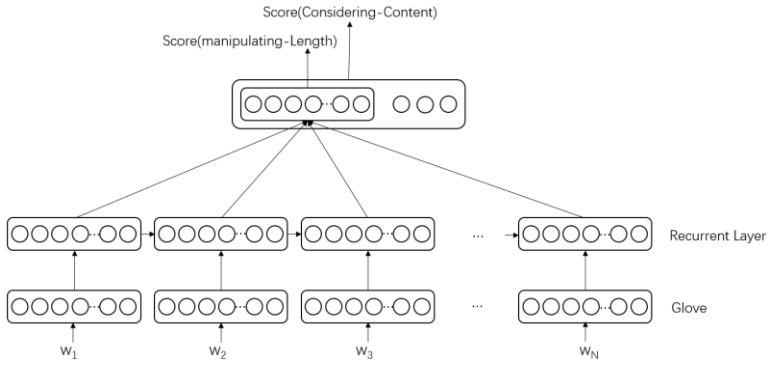

Figure 5: Architecture of the models we used.

Analysis

First, we trained the 'Manipulating-Length-GRU' model (see Table 4 and Table 5).

Table 4: Performance of 'Manipulating-Length-GRU' model for prompts 1-6.

Data	Prompt					
	1	2	3	4	5	6
Ori	**83.7**	69.5	68.7	80.1	**81.5**	80.2
Ori + Ch	83.6	69.5	68.8	80.4	81.4	80.5
Ori + Fr	83.5	**69.9**	**69.0**	**80.5**	**81.5**	**80.7**

Table 4 shows the performance of the 'Manipulating-Length-GRU' model for prompts with a small score range. 'Ori' means original data. 'Ori + Ch' means augmented data with back-translation essays using Chinese. The same goes for 'Ori + Fr'. The scores of back-translation essays are the same as those of the original essays. Table 5 shows the performance of the 'Manipulating-Length-GRU' model for prompts 7, 8 with a large score range. For the augmented data using adjusted scores, we marked the score adjustment Equation number and the value of variable v in the Equation next to the data. For example, [(4), v=2] means giving all back-translation essays in the prompt 2 points higher scores than the original essay scores. The value marked with '*' is slightly bigger than the value marked with '^'.

Table 5: Performance of 'Manipulating-Length-GRU' model for prompts 7 and 8.

Data	Prompt	
	7	8
Ori	**80.7**	70.5
Ori + Ch	80.2	70.4
Ori + Ch [(4), v=1]	80.0	70.3
Ori + Ch [(5), v=1]	79.7	**70.7**^
Ori + Ch [(4), v=2]	-	69.7
Ori + Ch [(6)]	80.3	70.6
Ori + Ch [(6), (12)]	-	**70.7***
Ori + Fr	80.1	69.9
Ori + Fr [(4), v=1]	80.1	70.2
Ori + Fr [(5), v=1]	79.4	69.8
Ori + Fr [(4), v=2]	-	69.7
Ori + Fr [(6)]	80.4	70.4

For prompts 2, 3, 4 and 6, 'Ori + Fr' showed the best performance. Except for the 'Ori + Ch' for prompt 5, for prompts 2 to 6, the performance was improved by using back-translation essays and original scores. For prompt 1, the performance did not improve, and we suspect that this is because prompt 1 has a relatively larger score range than prompts 2 to 6. For prompt 8, since the score range is twice that of prompt 7, [(4), v=2] was also applied. Except for the 'Ori + Ch [(5), v=1]' for prompt 8, if all essays of the prompt are scored using Equation 4 or Equation 5, the performance is lower than that when the score is not adjusted. In contrast, the augmented data using Equation 6 showed a higher performance than when using original scores. For prompt 8, the augmented data improved the performance compared to the original data. For prompt 7, the performance of the original data is the best.

For prompt 8,'Ori + Ch [(5), v=1]' performed better than Ori + Ch [(6)]. We defined Equation 12 as follows and augmented the data by using Equations 6 and 12:

$$s_{new}^{d} = \max\left(0, s_{ori}^{d} - 1\right), d \in \{e | s_{ori}^{e} \leq 40 \cap e \in E_8\} \quad (12)$$

The performance of the new augmented data was slightly higher than that of 'Ori + Ch [(5), v=1]'. This indicates that Equation 6 is still effective. Ori + Ch [(6)] performed worse than 'Ori + Ch

[(5), v=1]' because the number of applied essays was smaller. Equation 6 was applied to 146 essays, as shown in Table 3, but Equation 5 was applied to a total of 723 essays.

For prompt 5, the performance is decreased when 'Ori + Ch' is applied, and for prompt 8, the performance is improved when 'Ori + Ch [(5), v=1]' is applied. We suspect that some information is lost when translating low-score essays using a multibyte language.

Figure 6 demonstrates QWK score on the validation set for every epoch after training the model once. The performance in the graph has a certain difference from the final performance. The performance converges until an epoch number below 10, and the convergence speed is faster when the augmented data is used. However, there are cases where the performance improves even for epochs after 10, especially for prompts 7 and 8.

Using the augmented data is twice as long as using the original data. To reduce the training time, we attempted to reduce the number of epochs for augmented data. After obtaining the result by setting the number of epochs to 50, we determined the first epoch number with the best performance (see Figure 7). When using the augmented data, the first epoch number of the best model was much smaller than when using the original data. This implies that using augmented data gets to the best model faster though increasing the training time for one epoch. Therefore, when training the next model, we set the number of epochs to 30 for augmented data. The training time of the augmented data is 1.2 (2 * 3/5) times longer than that of the original data.

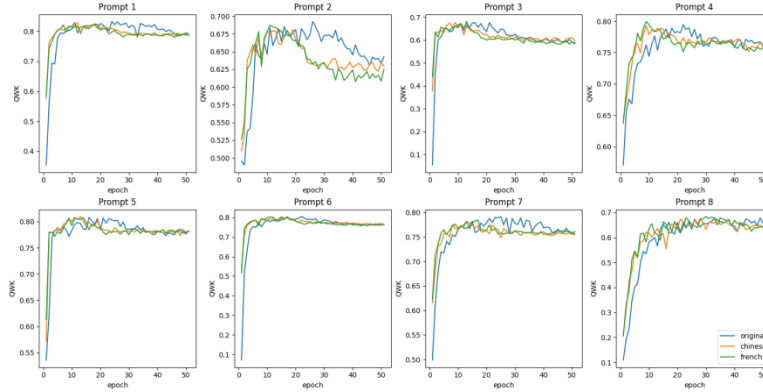

Figure 6: QWK score on the validation set.

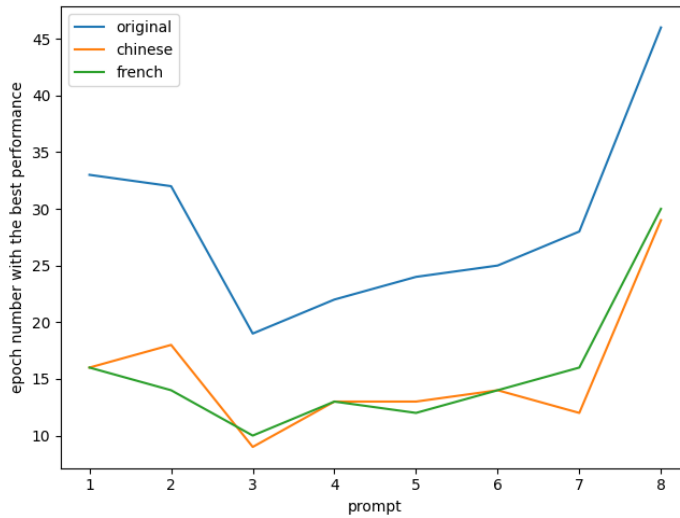

Figure 7: The first epoch number with the best performance for all prompts.

Second, we trained the 'Considering-Content-LSTM' model (see Table 6). In Table 6, the average improvement value was obtained by dividing the improvement value for 5 prompts by 8. The value marked with '*' is slightly bigger than the value marked with '^'.

Table 7 shows a performance comparison between our evaluation model and previous models. Those models used the

same cross-validation data to evaluate. In fact, models in [11] outperform these previous models without data augmentation.

Table 6: Performance of 'Considering-Content-LSTM' model for prompts 2, 3, 4, 6, and 8.

Data	Prompt								Average Improvement
	1	2	3	4	5	6	7	8	
Ori	**84.1**	71.0	69.5	80.1	82.6	81.3	**80.7**	71.0	-
Ori + Ch [(6), (12)]	-	71.2	**69.7**	80.3	-	81.5	-	**71.7**	0.2^
Ori + Fr [(6)]	-	**71.3**	69.6	**80.5**	-	**81.8**	-	71.4	0.2*

Table 7: Performance comparison.

Data	Prompt								Average
	1	2	3	4	5	6	7	8	
Ori[8]	82.1	68.8	69.4	80.5	80.7	**81.9**	**80.8**	64.4	76.1
Ori[10]	82.2	68.2	67.2	**81.4**	80.3	81.1	80.1	70.5	76.4
Augmented	**84.1**	**71.3**	**69.7**	80.5	**82.6**	81.8	80.7	**71.7**	77.8

Through the experiment on the first model, for prompts 2, 3, 4, 6, and 8, the effectiveness of the augmented data was confirmed. New results were obtained using augmented data in the second model. The performance was also improved in the second model.

The performance was improved by 0.2% on average for both models using the augmented data.

Conclusions

In this paper, data augmentation was first introduced in AES and the score was adjusted in consideration of the characteristics of the AES task. We improved the performance of AES by using back-translation essays and adjusted scores. We generated back-translation essays and adjust scores for the ASAP dataset, and confirmed the effectiveness of the augmented data. We used different score adjustment methods for specific prompts to find a reasonable method.

We generated back-translation essays for the ASAP dataset using Chinese and French. It was effective to maintain the score as it was for prompts with a small score range. For prompts with a large score range, based on the highest frequency score, it was effective to increase the score for the high-score essays and maintain the score for the low-score essays. For prompts 2, 3, 4, 6, and 8, higher performance was obtained than when using the original data. The performance was improved by 0.2% on average. In addition, we found that the augmented data gets to the best model faster than the original data, reducing the effect of increasing time from data augmentation to some extent.

By improving the performance of AES using data augmentation, it is possible to further improve the performance to a certain extent even when the dataset cannot be sufficiently established due to various limitations. In other words, it provided new research possibilities for the AES task.

Currently, our research has the following limitations. Score adjustment was not performed more mathematically. The experiment did not proceed sufficiently. Comparative models and datasets used for evaluation are insufficient.

In the future, we will explore more efficient, more mathematically theoretical, and practical score adjustment methods for back-translation essays. In addition, the present method will be applied to other datasets. We also plan to research other data augmentation techniques and their corresponding score adjustment methods for the AES task.

References

1. B Alhayani, HJ Mohammed, IZ Chaloob, JS Ahmed. Effectiveness of artificial intelligence techniques against cyber security risks apply of IT industry. Materials Today: Proceedings. 2021.
2. S Huang, J Yang, S Fong, Q Zhao. Artificial intelligence in the diagnosis of COVID-19: Challenges and perspectives. International Journal of Biological Sciences. 2021; 17: 1581-1587.

3. Junwei Ma, Yankun Wang, Xiaoxu Niu, Sheng Jiang, Zhiyang Liu. A comparative study of mutual information-based input variable selection strategies for the displacement prediction of seepage-driven landslides using optimized support vector regression. Stochastic Environmental Research and Risk Assessment. 2022; 1-21.
4. Junrong Zhang, Huiming Tang, Dwayne D Tannant, Chengyuan Lin, Ding Xia, et al. Combined forecasting model with CEEMD-LCSS reconstruction and the ABC-SVR method for landslide displacement prediction. Journal of Cleaner Production. 2021; 293.
5. JL Elman. Finding structure in time. Cognitive science 14.2. 1990; 179-211.
6. ZL Yang, ZH Dai, YM Yang. Xlnet: generalized autoregressive pretraining for language understanding. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Vancouver, Canada. 2019; 32: 5754–5764.
7. EB Page. The use of the computer in analyzing student essays. International review of education. 1968; 210-225.
8. K Taghipour, HT Ng. A neural approach to automated essay scoring, Proceedings of the 2016 conference on empirical methods in natural language processing. 2016; 1882-1891.
9. D Alikaniotis, H Yannakoudakis, M Rei. Automatic text scoring using neural networks, arXiv preprint arXiv:1606.04289. 2016.
10. F Dong, Y Zhang, J Yang. Attention-based recurrent convolutional neural network for automatic essay scoring. Proceedings of the 21st conference on computational natural language learning (CoNLL 2017). 2017; 153-162.
11. S Jeon, M Strube. Countering the Influence of Essay Length in Neural Essay Scoring. Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing. 2021; 32-38.
12. Y Yang, J Zhong. Automated essay scoring via example-based learning. International Conference on Web Engineering. Cham: Springer. 2021; 201-208.
13. J Xue, X Tang, L Zheng. A Hierarchical BERT-Based Transfer Learning Approach for Multi-Dimensional Essay Scoring. IEEE Access. 2021; 9: 125403-125415.

14. J Liu, Y Xu, Y Zhu. Automated essay scoring based on two-stage learning, arXiv preprint arXiv:1901.07744. 2019.
15. S Hochreiter, J Schmidhuber. Long short-term memory. Neural Computation. 1997; 9: 1735–1780.
16. K Cho, B Van Merrienboer, C Gulcehre. Learning ¨ phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. 2014.
17. J Devlin, MW Chang, K Lee. Bert: pre-training of deep bidirectional transformers for language understanding. 2018. Available online at: https://arxiv.org/abs/1810.04805.
18. WY Zou, R Socher, D Cer. Bilingual word embeddings for phrase-based machine translation. Proceedings of the 2013 conference on empirical methods in natural language processing. 2013; 1393-1398.
19. R Collobert, J Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. Proceedings of the 25th international conference on Machine learning. 2008; 160-167.
20. R Collobert, J Weston, L Bottou. Natural language processing (almost) from scratch. Journal of machine learning research. 2011; 12: 2493-2537.
21. J Pennington, R Socher, CD Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014; 1532-1543.
22. X Zhang, Xiang, J Zhao, Y LeCun. Character-level convolutional networks for text classification. Advances in neural information processing systems. 2015; 28: 649-657.
23. WY Wang, D Yang. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. Proceedings of the 2015 conference on empirical methods in natural language processing. 2015; 2557-2563.
24. AW Yu, D Dohan, MT Luong. Qanet: Combining local convolution with global self-attention for reading comprehension. 2018.
25. J Lun, J Zhu, Y Tang. Multiple data augmentation strategies for improving performance on automatic short answer

scoring. Proceedings of the AAAI Conference on Artificial Intelligence. 2020; 34: 13389-13396.
26. Q Xie, Z Dai, E Hovy. Unsupervised data augmentation for consistency training. Advances in Neural Information Processing Systems (NeurIPS 2020). 2020; 33: 6256-6268.
27. C Coulombe. Text data augmentation made simple by leveraging nlp cloud apis. 2018.
28. M Regina, M Meyer, S Goutal. Text Data Augmentation: Towards better detection of spear-phishing emails. 2020.
29. D Ramesh, SK Sanampudi. An automated essay scoring systems: a systematic literature review. Artificial Intelligence Review. 2021; 1-33.
30. YJ Jong, YJ Kim, OC Ri. Improving Performance of Automated Essay Scoring by using back-translation essays and adjusted scores. 2022.