On the Effectiveness of Curriculum Learning in Educational Text Scoring

Zijie Zeng, Dragan Gašević and Guanliang Chen*

Centre for Learning Analytics, Monash University, Australia
{Zijie.Zeng, Dragan.Gasevic, Guanliang.Chen}@monash.edu

Abstract

Automatic Text Scoring (ATS) is a widely-investigated task in education. Existing approaches often stressed the structure design of an ATS model and neglected the training process of the model. Considering the difficult nature of this task, we argued that the performance of an ATS model could be potentially boosted by carefully selecting data of varying complexities in the training process. Therefore, we aimed to investigate the effectiveness of *curriculum learning* (CL) in scoring educational text. Specifically, we designed two types of difficulty measurers: (i) *pre-defined*, calculated by measuring a sample's readability, length, the number of grammatical errors or unique words it contains; and (ii) *automatic*, calculated based on whether a model in a training epoch can accurately score the samples. These measurers were tested in both the *easy-to-hard* to *hard-to-easy* training paradigms. Through extensive evaluations on two widely-used datasets (one for short answer scoring and the other for long essay scoring), we demonstrated that (a) CL indeed could boost the performance of state-of-the-art ATS models, and the maximum improvement could be up to 4.5%, but most improvements were achieved when assessing short and easy answers; (b) the pre-defined measurer calculated based on the number of grammatical errors contained in a text sample tended to outperform the other difficulty measurers across different training paradigms.

Introduction

Automatic Text Scoring (ATS) is a common but important task in the field of education. With the aid of ATS techniques, instructors can automatically assess the quality of student-authored texts such as short answers to open-ended questions (Sung et al. 2019; Lun et al. 2020; Leacock and Chodorow 2003; Xia et al. 2020; Ramachandran, Cheng, and Foltz 2015) and relatively longer essays (Uto, Xie, and Ueno 2020; Burstein and Chodorow 1999; Attali and Burstein 2006; Rodriguez, Jafari, and Ormerod 2019; Amorim, Cançado, and Veloso 2018). Considering the large student-teacher ratio in certain educational scenarios, e.g., the ratio can be up to 10,000:1 or even worse in Massive Open Online Courses (Pappano 2012), ATS or writing assessment tools building upon ATS (e.g., AcaWriter (Knight et al. 2020) and Grammarly (Karyuatry 2018)) have been increasingly used in practice to facilitate instructors' teaching practices.

Given the important role of ATS in education, researchers have made great efforts in designing effective scoring algorithms with the aid of different techniques, such as early rule-based methods (Leacock and Chodorow 2003), subsequent machine learning methods with hand-crafted features as input (Mohler, Bunescu, and Mihalcea 2011; Sultan, Salazar, and Sumner 2016; Amorim, Cançado, and Veloso 2018), and recent methods based on deep neural networks that can automatically engineer features from input text (Xia et al. 2020; Rodriguez, Jafari, and Ormerod 2019; Sung, Dhamecha, and Mukhi 2019; Sung et al. 2019; Lun et al. 2020; Uto, Xie, and Ueno 2020; Ormerod, Malhotra, and Jafari 2021). In certain writing evaluation tasks, e.g., when assessing students' responses to the prompt questions, these ATS algorithms have demonstrated scoring performance comparable to human graders. However, there still exist scenarios which call for more research efforts to further improve the performance of ATS (Uto 2021; Ridley et al. 2020).

It is worth noting that most of the existing ATS studies boost the scoring performance by designing a more dedicated (and oftentimes more complicated) model structure to capture the unique characteristics of a writing assessment task. However, given the difficult nature of this task, i.e., sometimes even experienced human graders can disagree on the score assigned to a piece of writing (refer to Figure 1 to see such examples from the short answer scoring dataset [1] and the essay scoring dataset [2] released in the Kaggle platform), we argued that, in addition to designing more dedicated model structure, the process of the model training is worthy of our attention as well and this is where *curriculum learning* (CL) can potentially help. CL is a strategy used to train a prediction model by inputting data sorted in an easy-to-hard manner, which imitates the learning order in human curricula. As a model-agnostic training strategy, CL has been widely investigated and applied in various predictive tasks (Wu, Dyer, and Neyshabur 2020; Platanios et al. 2019) in terms of improving a model's generalization ability

*Corresponding author

[1] https://www.kaggle.com/competitions/asap-sas/

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

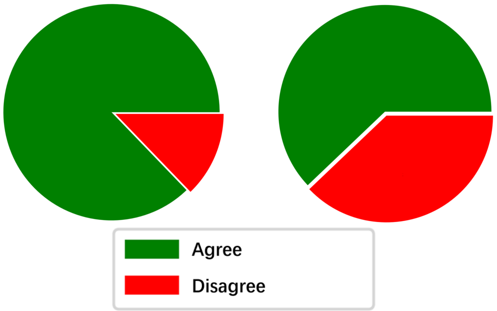

Figure 1: In the short-answer scoring dataset (left subfigure), there are 12.8% answers which received different scores from two independent experienced human graders, which were indicated as Disagree in red color; this number is even up to 37.9% in another dataset used for automatic essay scoring (right subfigure), which is essentially a more challenging task. We argued that Curriculum Learning can be beneficial in such scenarios.

and subsequently producing better prediction performance. Inspired by Wang, Chen, and Zhu (2021), which indicated that CL can be highly effective in enhancing a supervised prediction model when dealing with a difficult task (e.g., the automatic scoring of student-authored responses), we aimed to investigate how CL can be tapped to improve the performance of ATS in education. Formally, our study was guided by the following **R**esearch **Q**uestion:

RQ To what extent can curriculum learning strategies boost the performance of ATS methods used in education?

To answer the above question, we centered our work on the design of the two key components of a CL strategy (Wang, Chen, and Zhu 2021; Liu et al. 2018): (i) *difficulty measurer*, which determines the relative difficulty level of a training data sample; and (ii) *training scheduler*, which determines the data subset that should be input to a model in a specific training epoch based on the evaluation from the difficulty measurer. Inspired by previous works on proposing effective CL strategies in the broader NLP research (e.g., Spelling Error Correction (Gan, Xu, and Zan 2021) and Natural Answer Generation (Liu et al. 2018)) as well as the works on automatically characterizing textual data in education, we devised two types of CL strategies in this work, i.e., *pre-defined* and *automatic*, which are grouped according to whether any or both of the two key components described above are pre-defined by human experts or automatically learned in a data-driven fashion. It should be noted that these naming terminologies are in line with those summarized by Wang, Chen, and Zhu (2021). Specifically, we studied a total of four pre-defined CL strategies, in which the difficulty level of a piece of written text can be measured via calculating its length, readability, the number of grammatical errors or unique words it contains, and the training scheduler is defined as the linear continuous schedulers (Wang, Chen, and Zhu 2021). As documented in relevant CL studies in computer vision and NLP, in addition to presenting the training data in an easy-to-hard fashion, sometimes a model can achieve better prediction performance by reverting the training order to hard-to-easy (denoted as *anti-curriculum*). As there lacked prior studies on applying CL to tackle the task of AST and it remained largely unknown which training paradigm would benefit the most, we included both the easy-to-hard and hard-to-easy training paradigms to measure the effectiveness of the four CL strategies described above. Through extensive evaluations on two widely-used educational datasets, i.e., one for Automatic Short Answer Scoring (ASAS) and the other for Automatic Essay Scoring (AES), our work demonstrated that: (i) with the aid of CL, the performance of state-of-the-art ATS models can be further boosted with a maximum 4.5% improvement (measured by Quadratic Weighted Kappa); (ii) among the four investigated pre-defined difficulty measurers, the number of grammatical errors tended to give the most robust performance in measuring sample difficulty; (iii) no significant difference was observed between the pre-defined and automatic CL strategies, or between the easy-to-hard and hard-to-easy training paradigms.

Related Work

Automatic Text Scoring in Education

In education, accurate assessment of textual responses authored by students, along with timely and informative feedback carefully crafted by instructors, is an important task in helping students develop effective writing skills and improve their knowledge level (Dikli 2010). The completion of this task used to rely on manual efforts heavily. However, manual grading has often suffered from issues such as low precision, high inconsistency, and limited scalability (i.e., being unable to provide timely assessment to a large number of students' responses) (Fazal, Dillon, and Chang 2011; Valenti, Neri, and Cucchiarelli 2003). As a remedy, ATS has been proposed and widely investigated by researchers to facilitate instructors to perform scoring practices (Alikaniotis, Yannakoudakis, and Rei 2016; Ke and Ng 2019) and it is often used to assess students' responses to short-answer questions and essay prompts, which are denoted as ASAS and AES, respectively.

Broadly speaking, most of the existing ATS can be categorized into two categories (Bonthu, Rama Sree, and Krishna Prasad 2021; Uto, Xie, and Ueno 2020). One is often built upon traditional machine learning techniques e.g., Linear Regression (Nau, Haendchen Filho, and Passero 2017), Support Vector Machine (Gleize and Grau 2013), and Random Forests (Ishioka and Kameda 2017), whose performance is heavily dependent on the availability and quality of hand-crafted features such as the number of words contained in an answer (Platanios et al. 2019) and the number of distinct words in the answer (Li et al. 2016). The other is empowered by the recent deep learning techniques such as Bi-LSTM (Kim, Vizitei, and Ganapathi 2018) and BERT (Sung, Dhamecha, and Mukhi 2019), which can directly transform the raw text input as embedding-based representations to generate an assessment score without the need of manual feature engineering. For instance, driven by the great success achieved by pre-trained language models in various NLP tasks, Sung, Dhamecha, and Mukhi (2019) proposed to couple BERT (Devlin et al. 2019) with a single classification layer and fine-tuned the whole model on a labeled dataset for ASAS, whose scoring performance was up to 0.91 measured by weighted average F1. Though certain methods, e.g., those proposed by Taghipour and Ng (2016); Sung, Dhamecha, and Mukhi (2019), have demonstrated human-like scoring performance, there still exist scenarios in which further research efforts are needed to develop more accurate ATS techniques, e.g., the cross-prompt scenario in which data from auxiliary prompts is used to trained ATS models for the target prompt(Uto 2021; Ridley et al. 2020).

Noticeably, the ATS models described in the above studies, especially those based on state-of-the-art deep learning techniques, often treated the design of more dedicated model structures as their main means to boost the performance of ATS, while little attention has been given to the training process of these well-designed models. As explained before, considering the difficult nature of ATS in education, we argued that it might be worthwhile to apply CL to optimize the training process and assist an ATS model to reach its full potential.

Curriculum Learning

Curriculum learning (CL) refers to the strategy used to train a prediction model by imitating the meaningful learning order in human curricula, i.e., presenting the training samples in an easy-to-hard manner so as to enable the model to first optimize an easier version of the target problem and gradually consider harder versions, until solving the full target task of interest (Bengio et al. 2009). As indicated before, most of the existing CL strategies consist of two key components, i.e., *difficulty measurer* and *training scheduler* (Wang, Chen, and Zhu 2021) (or *scoring function* and *pacing function* in other relevant literature). Depending on whether any (or both) of the two components are designed with the aid of human expertise (or data-driven approaches), a strategy can be classified as either *pre-defined* or *automatic* CL. Take difficulty measurer as an example, Platanios et al. (2019) developed a pre-defined strategy in which the difficulty of input text was determined by using its length as a proxy (i.e., the longer the input text, the higher difficulty it has), while Gan, Xu, and Zan (2021) proposed an automatic strategy in which the difficulty was measured by calculating its training loss in a specific epoch.

Since its inception Bengio et al. (2009), CL has been demonstrated effective in boosting performance of various models in the research of computer vision and NLP (Soviany et al. 2020; Spitkovsky, Alshawi, and Jurafsky 2010; Tudor Ionescu et al. 2016; Gan, Xu, and Zan 2021; Platanios et al. 2019; Wei et al. 2016; Liu et al. 2018). For instance, when training on imbalance-distributed image data, the Dynamic Curriculum Learning framework proposed by Wang et al. (2019) employed a two-level curriculum schedulers, which consist of a dynamic sampling scheduler that adjusts the data distribution at each time step and balances the importance between the classification loss and the metric learning loss. In a different vein, when performing the task of natural answer generation, Liu et al. (2018) measured text difficulty from the perspective of Grammar (Stanford Parser score[3]) and trained the model first on the simple and low-quality data and then on the complex and high-quality data to gradually learn to generate reliable answers for questions of different complexity, outperforming the state-of-the-art by an average improvement of about 7.5% in terms of accuracy. More worthy of our attention is that, as indicated by Wang, Chen, and Zhu (2021), CL can be particularly useful when dealing with difficult tasks, e.g., those involving the use of higher-order cognitive skills to develop solutions, e.g., the task of assessing student-author responses investigated in our study.

It should be pointed out that, though most of the existing CL studies posited that a model's performance can be boosted to the maximum degree by adopting the easy-to-hard learning order, there have been some studies (Zhang et al. 2018; Pi et al. 2016; Braun, Neil, and Liu 2017) which demonstrated that, in certain cases, the model could be trained in an opposite learning order, i.e., from harder data to easier data (also called anti-curriculum learning (Wang, Chen, and Zhu 2021). For instance, Zhang et al. (2018) demonstrated that the hard-to-easy order, compared to its easy-to-hard counterpart, could lead to better model performance in neural machine translation.

Though CL has been demonstrated effective, few studies attempted to investigate its effectiveness in enhancing the assessment of textual responses authored by students in education, which is often deemed as a challenging and high-stake task (Gierl et al. 2014; Beseiso and Alzahrani 2020; Cao et al. 2020), motivating us to design effective CL strategies to further enhance the performance of existing ATS models.

Methods

Tasks and Datasets

Our study was centered around two common writing assessment tasks in education, i.e., ASAS and AES. Generally speaking, as the text length of an essay is often much longer than a short answer, AES is often regarded as a more challenging task than ASAS. The two datasets we used were released by the Hewlett Foundation to spur the development of novel techniques to tackle the two tasks described above, respectively.

Dataset for ASAS[4]. The dataset consists of about 17,000 answers written by students who were mainly from Grade 10 in the US as responses to 10 different prompt questions of subjects including Science, Biology, English, etc. The number of words contained in an answer ranges from 1 to 344, with an average of 41.7. It is noteworthy that each answer was assessed by two independent human graders. The scores given by the first grader are the ground truth that an ATS model should aim to predict, while the scores given by the second grader are only used to measure the agreement between different human graders. Notice that there are about 12.8% answers which received different scores from the two graders. Statistics of this dataset can be found in Table 1.

[3] http://nlp.stanford.edu/software/parser-faq.shtml

[4] https://www.kaggle.com/competitions/asap-sas/

Prompt	Subject Area	#Answers	Score Range	Inter-rater Agreement	Average Length
1	Science	1672	0-3	0.950	47.1
2	Science	1278	0-3	0.900	59.2
3	ELA	1808	0-2	0.681	47.7
4	ELA	1657	0-2	0.683	40.2
5	Biology	1795	0-3	0.962	25.1
6	Biology	1797	0-3	0.952	23.4
7	English	1799	0-2	0.959	41.1
8	English	1799	0-2	0.866	53
9	English	1798	0-2	0.782	49.7
10	Science	1640	0-2	0.887	41.4

Table 1: Statistics of the ASAS dataset. Note that ELA is short for English Language Arts. The inter-rater agreement was measured as the quadratic weighted kappa between scorer1 and scorer2.

Dataset for AES[5] The dataset consists of about $13,000$ essays written by students who were from Grade 7 to Grade 10 in the US as responses to eight different prompts. Similar to the ASAS dataset, each answer in this dataset was assessed by at least two human graders. The difference lies in that the final score to be predicted by a model was determined based on the scores provided by all human graders. Notice that there are 37.9% answers which received different scores from the human graders. Statistics of this dataset can be found in Table 2.

Prompt	Essay Type	#Essays	Score Range	Inter-rater Agreement	Average Length
1	PNE	1783	2-12	0.721	350
2	PNE	1800	1-6	0.814	350
3	SDR	1726	0-3	0.769	150
4	SDR	1770	0-3	0.851	150
5	SDR	1805	0-4	0.753	150
6	SDR	1800	0-4	0.776	150
7	PNE	1568	0-30	0.721	250
8	PNE	721	0-60	0.629	650

Table 2: Statistics of the AES dataset. Note that PNE is short for Persuasive / Narrative / Expository and SDR is short for Source Dependent Responses. The inter-rater agreement was measured as the quadratic weighted kappa between scorer1 and scorer2.

Models

Recall that our goal was to investigate the effectiveness of CL in boosting the performance of ATS models. To measure the capabilities of various CL strategies, we selected state-of-the-art models used for ASAS and AES as the testbeds in this study, as described below.

Model for ASAS. In line with previous studies (Xia et al. 2020; Sung, Dhamecha, and Mukhi 2019), given the limited number of scores that can be assigned to an answer (e.g., ranging from 0 to 3), we tackled the task of ASAS as a classification problem. We followed the approaches developed by (Sung, Dhamecha, and Mukhi 2019; Sung et al. 2019; Lun et al. 2020), which coupled BERT with a single classification layer as the scoring model and fine-tuned the whole model for each of the prompt question so as to enable the model to capture the task-specific characteristics and subsequently optimize the scoring performance.

Model for AES. Here, we adopted the approach developed by (Uto, Xie, and Ueno 2020), i.e., augmenting the BERT-based grader (i.e., the one used for ASAS described above) with human-crafted features as input to maximize the scoring performance. Particularly, the features were engineered on the essay level, and the rationale behind this is that, compared to a short answer, an essay is often much longer and can have a more complex structure, which may pose challenges to the BERT model to derive an accurate representation for the essay. By adding essay-level features as input, the BERT-based grader was expected to model the quality of an essay better. This approach has been reported to achieve state-of-the-art performance in AES (Ormerod, Malhotra, and Jafari 2021). Similar to ASAS, we built one AES for each of the prompts contained in the dataset.

Curriculum Learning Design

Difficulty measurer We investigated two types of difficulty measurers, i.e., *pre-defined* and *automatic*, as describe below.

Pre-defined. A key characteristic of pre-defined strategies lies in that the measurement of a training sample's difficulty often relies on human expertise (Wang, Chen, and Zhu 2021). Inspired by relevant studies on analyzing textual data in education (e.g., those characterizing the utterances generated by instructors in online one-on-one tutoring (Lin et al. 2022a,b) or analyzing students' posts made in discussion forums (Sha et al. 2021)), we designed a total of four difficulty measures for pre-defined CL strategies, as described below:

- `Length`, which measures the length of a piece of text as a proxy to its difficulty level (Platanios et al. 2019; Spitkovsky, Alshawi, and Jurafsky 2010). Here, the longer a response is, the more difficult it is considered to be.
- `Distinct-1`, which counts the number of unique words contained in a response to measure its difficulty level (Li et al. 2016). Here, the more unique words a response has, the more difficult it is considered to be. We acknowledged that longer text might contain more unique words. Thus, we scaled the number of unique words by the length of the textual response as the final difficulty measurer.
- `Readability`, which calculates the Flesch Reading Ease score (Farr, Jenkins, and Paterson 1951) of a response to measure its difficulty level. A Flesch Reading Ease score is of the range $[0, 100]$, with 0 representing being extremely difficult to read and 100 being extremely easy to read. Therefore, in this measurer, the lower the readability score, the more difficult the response is.

[5] https://www.kaggle.com/competitions/asap-aes/.

- `Errors`, which detects the number of grammatical errors and spelling mistakes contained in a response to measure its difficulty level. Here, the more errors a response contains, the more difficult it is considered to be. Similar to `Distinct-1`, we noticed that the longer the text, the more errors it might contain. We therefore scaled this measure by the text length as the difficulty measurer.

Automatic. Though pre-defined strategies have been demonstrated effective in various application scenarios, they are often plagued by their strong reliance on human expertise to define an appropriate difficulty measurer and an extensive search for effective combinations of difficulty measurer and training scheduler. Therefore, in addition to the four pre-defined CL strategies described above, we further designed an automatic difficulty measurer to dynamically select data samples based on instance-wise training loss and enable a more flexible training process. Specifically, the automatic difficulty measurer used in this study characterizes data samples as either *easy* and *difficult*, which represents the samples whose ground truth scores are *correctly* or *incorrectly* predicted by a model in a training epoch. Let p^t_{easy} and $p^t_{difficult}$ denote the probabilities of an individual *easy* sample and an individual *difficult* sample to be selected for model training at the current t-th epoch, we define r as the ratio between these two probabilities:

$$r = \frac{p^t_{difficult}}{p^t_{easy}}. \tag{1}$$

By choosing different values for r, we can enable the strategy to lay different emphasis on the easy and difficult samples. In particular, we explored two different ways to determine the value for r and consequently two variants of the automatic strategy:

- `Static`, which sets r to the same value across different training epochs. During experiments, r was empirically determined by searching in the range of $(0, 5]$ with an interval of 0.1. When $r < 1$ (i.e., $p^t_{difficult} < p^t_{easy}$), easy samples will be more likely to be selected for training; when $r > 1$, difficult samples will be more likely to be selected for training.
- `Adaptive`, the value of r at the current t-th epoch is based on the number of easy and difficult samples in the previous epoch. We denote the set of easy and difficult samples as E and D, respectively, and formally define:

  $$r = \frac{p^t_{difficult}}{p^t_{easy}} = \frac{|E|}{|D|}. \tag{2}$$

  Note that such a setting of r ensures that when there are relatively a larger portion of easy (or difficult) samples, the strategy tends to select difficult (or easy) samples more often for the subsequent training.

The sum of the sampling probabilities assigned to all training data should be equal to 1 in both cases for each epoch, e.g., it should be $p^t_{easy} * |E| + p^t_{difficult} * |D| = 1$ for the variant of `Adaptive`.

Training Scheduler

For both of the pre-defined and automatic difficulty measurers described above, we defined the training scheduler by using a function $\lambda(t)$ to map a training epoch number t to a scalar value $\lambda \in (0, 1]$, i.e., only the λ proportion of the easiest samples should be used to training a model at the t-th epoch. Here, the function is formally defined as:

$$\lambda(t) = \frac{t}{T}, \tag{3}$$

where T denotes the total number of epochs throughout the whole training process and $t \in [1, T]$. Essentially, this is a form of the linear continuous schedulers summarized by Wang, Chen, and Zhu (2021). We leave the exploration of other forms of training schedulers (e.g., root function (Platanios et al. 2019) and geometric progression function (Penha and Hauff 2020)) in our future studies.

As explained before, given the inconsistent findings of CL in various scenarios, we fed the training samples not only in an easy-to-hard order but also in a hard-to-easy order (denoted as `Anti-CL`) to evaluate the effectiveness of the four pre-defined difficulty measurers. That is, only the λ proportion of the most difficult samples should be used to train a model at the t-th epoch.

Experimental Setup

Feature engineering for the AES model. Following (Uto, Xie, and Ueno 2020), we engineered four types of essay-level features as part of the input to empower the AES model described in Sec., including length-based features, syntactic features, word-level features, and readability features. Note that each feature was standardized to have a mean of 0 and a standard deviation of 1.0.

Baselines. We implemented two baselines for comparisons: (i) `Baseline w/o CL`, which refer to the vanilla versions of the selected ASAS and AES models without applying any CL strategies; and (ii) `Random curriculum`, in which the proportion of samples used at the t-th training epoch is the same as that of a CL strategy (i.e., as defined in Equation (3)), but the samples were randomly selected from the whole dataset, i.e., being in random difficulty order. This was used to scrutinize whether the observed performance change of a model was caused due to the changing sample size in different epochs (Wu, Dyer, and Neyshabur 2020).

Model implementation. We constructed the scoring models based on the pre-trained *bert-base-cased* encoder[6] implemented by the Python package Transformers[7]. Specifically, to obtain the ASAS scoring model, we simply added a classification layer on top of the *bert-base-cased*. For the AES scoring model, the representation vector output by the *bert-base-cased* encoder and the essay-level features were concatenated to reach an augmented representation vector, which served as the input into a regression layer (linear layer with sigmoid activation) to predict the final score.

[6] It had 12 layers, with 768 neurons in each hidden layer and the number of attention heads is 12.

[7] https://github.com/huggingface/transformers

Different from the ASAS scoring model, the AES model training adopts the mean square error (MSE) loss function, where the scores of the training samples are normalized to $[0, 1]$ (rescaled to the original score range at the prediction stage). All the codes used in this study can be accessed via https://github.com/douglashiwo/CurriculumLearningATS.

Model training. For each prompt, we randomly split the data into training, validation, and testing sets in the ratio of $70\% : 15\% : 15\%$. When training a model, we set the batch-size as 16 and the number of training epochs as 5. We selected the learning rate from $\{2e-5, 3e-5, 5e-5\}$ and Adam with decoupled weight to optimize the model. Note that the above parameter selections were guided by Devlin et al. (2019). The best combinations of parameters for each prompt were determined based on the model's performance in the validation set (measured by QWK). Each reported result is a mean over 5 independent runs with the same hyperparameters.

Model evaluation. In line with previous works (Uto, Xie, and Ueno 2020; Ormerod, Malhotra, and Jafari 2021), we adopted the metric Quadratic Weighted Kappa (QWK) to measure the agreement between the predicted scores derived by an ATS model and the ground truth scores.

Results

Results on ASAS. Table 3 details the results on the ASAS dataset. We can observe that `Random curriculum` showed no improvement over `Baseline w/o CL` on average QWK, implying that simply changing the size of the training set over each epoch (time step) cannot guarantee performance improvement. By comparing the results of `Baseline w/o CL` with those of different CL strategies, we can make several interesting observations. Firstly, when considering the average QWK over all prompts, we noticed that both pre-defined and automatic measurers delivered better scoring performance. For instance, `Readability` and `Errors` measurers outperformed `Baseline w/o CL` in both of the `Curr` and `Anti-Curr` schedulers. Besides, both the two automatic strategies outperformed `Baseline w/o CL`. Among all these strategies, the `Static` gained the maximum improvement (4.5%) over `Baseline w/o CL`. Secondly, when scrutinizing the results in each prompt, interestingly, we found that both the pre-defined and automatic strategies seemed to have been effective in certain prompts (e.g., Prompt 2, 3 and 4). Notice that these prompts contained a much larger fraction of answers that can be difficult to be accurately assessed. For example, the fraction of answers which received different scores from human graders in Prompt 3 was 23.9%. Note that Prompt 3 was also the one in which CL strategies presented the greatest improvements over baseline. This suggests that, CL tended to be effective when dealing with challenging tasks, which is in line with the findings reported by Wang, Chen, and Zhu (2021). Thirdly, among the four pre-defined difficulty measurers, `Readability` tended to be superior to the others, i.e., achieving the best performance when employed with the `Anti-Curr` scheduler and the second best when employed with the `Curr` scheduler. Finally, the advantage of the `Curr` scheduler over the `Anti-Curr` scheduler was unclear, since these two schedulers had each shown some superiority in specific prompts. This is also supported by Zhang et al. (2018); Pi et al. (2016); Braun, Neil, and Liu (2017); Wang, Chen, and Zhu (2021), which demonstrated that the easy-to-hard training order is not necessarily better than the order of hard-to-easy, reminding us that a training sample perceived easy by human might not be as easy for machine learning models and vice versa. To summarize, these results together imply that CL brought performance improvement of certain extent in scoring relatively short and easy answers.

Results on AES. Table 4 details the results on the AES dataset. Based on Table 4, we can make several observations similar to the ASAS results presented in Table 3. Firstly, no overall improvement was brought by `Random curriculum` compared to `Baseline w/o CL`. This corroborates that it is necessary to consider the difficulty of training samples when applying CL strategies to ATS models. Secondly, both the pre-defined and automatic measurers displayed certain improvements over `Baseline w/o CL`. For instance, `Readability`, `Errors`, and `Distinct-1` all outperformed `Baseline w/o CL` with the `Anti-Curr` scheduler. As for the automatic strategies, only `Static` was shown to be superior to `Baseline w/o CL`. Among all these strategies, the best performance was given by the pre-defined measurer `Readability` with the `Anti-Curr` scheduler. Thirdly, among the four proposed measurers, `Errors` tended to be more robust compared to the others with both the `Curr` and `Anti-Curr` schedulers, though there is no significant superiority observed between `Curr` and `Anti-Curr`. However, it is worth noting that all the observed improvements are rather limited, i.e., less than 1%. Also, when delving into the results for each prompt, we notice that the proposed strategies tended to be more effective in certain prompts (i.e., Prompt 1, 3, 4, 6). Surprisingly, these prompts have a relatively higher fraction of essays (about 70% on average) which received the same score from human graders, while this fraction is only 48.3% for the rest of the prompts (i.e., Prompt 2, 5, 7, 8). Therefore, prompts 1,3,4 and 6 can be regarded as relatively easier to be assessed. This implies that CL might be rather limited in boosting the ATS performance when dealing with particularly difficult tasks. For instance, the fraction of essays in Prompt 7 and Prompt 8 which received the same score from human graders are only 29% and 28%, respectively. When scrutinizing the results on these two prompts, most of the proposed CL strategies showed no performance improvement over `Baseline w/o CL`.

Discussion and Conclusion

Automatic scoring of student-authored responses, e.g., short answers and essays, is a long-standing task in the field of education. Though various models have been proposed to tackle this task, it remained largely unknown whether the performance of these models could be further boosted by carefully selecting data in the training process. In this study, we therefore investigated the effectiveness of CL strategies

Prompt ID			1	2	3	4	5	6	7	8	9	10	Avg.
Bseline w/o CL			0.803	0.603	0.069	0.678	0.657	0.777	0.631	0.612	0.706	0.755	0.629
Random difficulty			**0.813**	**0.630**	**0.115**	**0.689**	**0.689**	0.696	0.604	0.574	**0.715**	0.740	0.627
Pre-defined	Curr	Readability	**0.818**	**0.653**	0.052	**0.700**	0.641	**0.789**	**0.648**	0.573	**0.720**	**0.768**	**0.636**
		Length	**0.806**	**0.654**	**0.149**	**0.692**	0.620	0.614	0.627	0.583	**0.718**	0.710	0.618
		Errors	**0.817**	**0.618**	**0.092**	**0.698**	**0.732**	0.747	0.615	0.597	0.701	0.725	**0.634**
		Distinct-1	0.799	**0.636**	**0.126**	**0.718**	**0.666**	**0.782**	**0.641**	0.577	**0.712**	0.710	**0.637**
	Anti-Curr	Readability	0.786	**0.657**	**0.197**	**0.701**	**0.741**	0.685	0.594	0.585	**0.722**	0.741	**0.641**
		Length	0.766	0.601	0.045	**0.706**	**0.723**	0.761	0.616	0.546	**0.726**	0.694	0.618
		Errors	0.766	0.598	**0.167**	0.677	**0.680**	0.708	**0.672**	0.597	0.696	0.754	**0.631**
		Distinct-1	**0.811**	**0.677**	**0.121**	**0.699**	0.628	0.672	**0.649**	0.576	0.706	0.750	0.629
Automatic		Static	**0.826**	**0.664**	**0.208**	**0.698**	**0.712**	**0.821**	**0.633**	0.588	0.688	0.730	**0.657**
		Adaptive	0.764	**0.673**	**0.182**	**0.682**	**0.733**	0.776	0.597	0.575	0.691	0.733	**0.641**

Table 3: Results (QWK) on the ASAS dataset. Results in bold indicate being superior to that of Baseline w/o CL. Underlined results indicate being superior to that of Random Diffiuclty. Curr and Anti-Curr denote the easy-to-hard and hard-to-easy learning orders, respectively. A higher QWK indicates a better model performance.

Prompt ID			1	2	3	4	5	6	7	8	Avg.
Bseline w/o CL			0.748	0.618	0.628	0.802	0.788	0.793	0.823	0.677	0.735
Random difficulty			**0.770**	0.585	**0.659**	**0.813**	0.769	0.789	0.771	0.669	0.728
Pre-defined	Curr.	Readability	**0.761**	**0.627**	**0.632**	0.792	0.775	**0.798**	0.820	0.617	0.729
		Length	**0.765**	0.614	**0.698**	**0.815**	**0.795**	0.787	0.818	0.577	0.734
		Errors	**0.768**	0.616	**0.645**	**0.832**	0.769	0.778	0.811	0.670	**0.736**
		Distinct-1	**0.781**	**0.644**	**0.639**	**0.803**	0.777	0.775	0.806	0.628	0.731
	Anti-Curr.	Readability	0.742	0.615	**0.698**	**0.820**	**0.790**	**0.810**	0.812	0.658	**0.743**
		Length	0.745	0.575	**0.673**	**0.807**	**0.799**	**0.804**	0.819	0.631	0.732
		Errors	**0.788**	0.563	**0.663**	**0.806**	0.758	**0.806**	0.818	**0.683**	**0.736**
		Distinct-1	**0.777**	0.579	**0.675**	**0.814**	0.780	**0.804**	**0.833**	0.638	**0.737**
Automatic		Static	**0.779**	**0.639**	**0.685**	0.801	**0.790**	0.790	0.818	0.621	**0.740**
		Adaptive	**0.800**	0.616	**0.656**	0.793	0.770	**0.800**	**0.824**	0.592	0.731

Table 4: Results (QWK) on the AES dataset. Results in bold indicate being superior to that of Baseline w/o CL. Underlined results indicate being superior to that of Random Diffiuclty. Curr and Anti-Curr denote the easy-to-hard and hard-to-easy learning orders, respectively. A higher QWK indicates a better model performance.

in empowering the performance of ATS in the tasks of ASAS and AES. Specifically, we designed a set of four pre-defined measurers and one automatic measurer to describe the difficulty of a data sample, and investigated their effectiveness in both the easy-to-hard and hard-to-easy training paradigms. Through extensive evaluations on two different datasets, we showed that CL indeed can boost the performance of existing state-of-the-art ATS models used in education and the number of grammatical errors contained in a textual response can be used as an effective metric to measure the training difficulty of the response. More importantly, we demonstrated that, when assessing relatively short and easy answers, CL tended to display a stronger power in empowering ATS models and the brought improvement can be up to 4.5% measured in QWK. However, when dealing with relatively longer and more challenging essays, CL showed little improvement compared to the baselines. To understand the reason behind this and further improve ATS models, one future direction to improve this study is to dissect and observe how those ATS models change during the training process (e.g., the attention weights given to the input text across different training epochs), based on which better CL strategies can be developed. In addition, as we only investigated one type of training scheduler in this study, it would be worthwhile to incorporate more advanced scheduling strategies (e.g., those proposed in Platanios et al. (2019); Penha and Hauff (2020)) to further empower the ATS models, especially for the task of AES.

References

Alikaniotis, D.; Yannakoudakis, H.; and Rei, M. 2016. Automatic Text Scoring Using Neural Networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 715–725.

Amorim, E.; Cançado, M.; and Veloso, A. 2018. Automated essay scoring in the presence of biased ratings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 229–237.

Attali, Y.; and Burstein, J. 2006. Automated essay scoring with e-rater® V. 2. *The Journal of Technology, Learning and Assessment*, 4(3).

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Beseiso, M.; and Alzahrani, S. 2020. An empirical analysis of BERT embedding for automated essay scoring. *Int. J. Adv. Comput. Sci. Appl.*, 11(10): 204–210.

Bonthu, S.; Rama Sree, S.; and Krishna Prasad, M. 2021. Automated Short Answer Grading Using Deep Learning: A Survey. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, 61–78. Springer.

Braun, S.; Neil, D.; and Liu, S.-C. 2017. A curriculum learning method for improved noise robustness in automatic speech recognition. In *2017 25th European Signal Processing Conference (EUSIPCO)*, 548–552. IEEE.

Burstein, J.; and Chodorow, M. 1999. Automated essay scoring for nonnative English speakers. In *Computer mediated language assessment and evaluation in natural language processing*.

Cao, Y.; Jin, H.; Wan, X.; and Yu, Z. 2020. Domain-adaptive neural automated essay scoring. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1011–1020.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Dikli, S. 2010. The nature of automated essay scoring feedback. *Calico Journal*, 28(1): 99–134.

Farr, J. N.; Jenkins, J. J.; and Paterson, D. G. 1951. Simplification of Flesch reading ease formula. *Journal of applied psychology*, 35(5): 333.

Fazal, A.; Dillon, T.; and Chang, E. 2011. Noise reduction in essay datasets for automated essay grading. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 484–493. Springer.

Gan, Z.; Xu, H.; and Zan, H. 2021. Self-Supervised Curriculum Learning for Spelling Error Correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3487–3494. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Gierl, M. J.; Latifi, S.; Lai, H.; Boulais, A.-P.; and De Champlain, A. 2014. Automated essay scoring and the future of educational assessment in medical education. *Medical education*, 48(10): 950–962.

Gleize, M.; and Grau, B. 2013. Limsiiles: Basic english substitution for student answer assessment at semeval 2013. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, 598–602.

Ishioka, T.; and Kameda, M. 2017. Overwritable automated Japanese short-answer scoring and support system. In *Proceedings of the International Conference on Web Intelligence*, 50–56.

Karyuatry, L. 2018. Grammarly as a tool to improve students' writing quality: Free online-proofreader across the boundaries. *JSSH (Jurnal Sains Sosial dan Humaniora)*, 2(1): 83–89.

Ke, Z.; and Ng, V. 2019. Automated Essay Scoring: A Survey of the State of the Art. In *IJCAI*, volume 19, 6300–6308.

Kim, B.-H.; Vizitei, E.; and Ganapathi, V. 2018. GritNet: Student Performance Prediction with Deep Learning. arXiv:1804.07405.

Knight, S.; Shibani, A.; Abel, S.; Gibson, A.; and Ryan, P. 2020. AcaWriter: A learning analytics tool for formative feedback on academic writing. *Journal of Writing Research*.

Leacock, C.; and Chodorow, M. 2003. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4): 389–405.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, W. B. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 110–119.

Lin, J.; Rakovic, M.; Lang, D.; Gasevic, D.; and Chen, G. 2022a. Exploring the Politeness of Instructional Strategies from Human-Human Online Tutoring Dialogues. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, 282–293.

Lin, J.; Singh, S.; Sha, L.; Tan, W.; Lang, D.; Gašević, D.; and Chen, G. 2022b. Is it a good move? Mining effective tutoring strategies from human–human tutorial dialogues. *Future Generation Computer Systems*, 127: 194–207.

Liu, C.; He, S.; Liu, K.; Zhao, J.; et al. 2018. Curriculum Learning for Natural Answer Generation. In *IJCAI*, 4223–4229.

Lun, J.; Zhu, J.; Tang, Y.; and Yang, M. 2020. Multiple data augmentation strategies for improving performance on automatic short answer scoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 13389–13396.

Mohler, M.; Bunescu, R.; and Mihalcea, R. 2011. Learning to Grade Short Answer Questions Using Semantic Similarity Measures and Dependency Graph Alignments. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 752–762.

Nau, J.; Haendchen Filho, A.; and Passero, G. 2017. Evaluating Semantic Analysis Methods For Short Answer Grading Using Linear Regression. *Sciences*, 3(2): 437–450.

Ormerod, C. M.; Malhotra, A.; and Jafari, A. 2021. Automated essay scoring using efficient transformer-based language models. arXiv:2102.13136.

Pappano, L. 2012. The Year of the MOOC. *The New York Times*, 2(12): 2012.

Penha, G.; and Hauff, C. 2020. Curriculum learning strategies for ir. In *European Conference on Information Retrieval*, 699–713. Springer.

Pi, T.; Li, X.; Zhang, Z.; Meng, D.; Wu, F.; Xiao, J.; and Zhuang, Y. 2016. Self-paced boost learning for classification. In *IJCAI*, 1932–1938.

Platanios, E. A.; Stretcu, O.; Neubig, G.; Póczos, B.; and Mitchell, T. M. 2019. Competence-based Curriculum Learning for Neural Machine Translation. In *NAACL-HLT*.

Ramachandran, L.; Cheng, J.; and Foltz, P. 2015. Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, 97–106.

Ridley, R.; He, L.; Dai, X.; Huang, S.; and Chen, J. 2020. Prompt Agnostic Essay Scorer: A Domain Generalization Approach to Cross-prompt Automated Essay Scoring. arXiv:2008.01441.

Rodriguez, P. U.; Jafari, A.; and Ormerod, C. M. 2019. Language Models and Automated Essay Scoring. arXiv:1909.09482.

Sha, L.; Rakovic, M.; Whitelock-Wainwright, A.; Carroll, D.; Yew, V. M.; Gasevic, D.; and Chen, G. 2021. Assessing Algorithmic Fairness in Automatic Classifiers of Educational Forum Posts. In *International conference on artificial intelligence in education*, 381–394. Springer.

Soviany, P.; Ardei, C.; Ionescu, R. T.; and Leordeanu, M. 2020. Image difficulty curriculum for generative adversarial networks (CuGAN). In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3463–3472.

Spitkovsky, V. I.; Alshawi, H.; and Jurafsky, D. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 751–759.

Sultan, M. A.; Salazar, C.; and Sumner, T. 2016. Fast and Easy Short Answer Grading with High Accuracy. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1070–1075.

Sung, C.; Dhamecha, T.; Saha, S.; Ma, T.; Reddy, V.; and Arora, R. 2019. Pre-training BERT on domain resources for short answer grading. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6071–6075.

Sung, C.; Dhamecha, T. I.; and Mukhi, N. 2019. Improving Short Answer Grading Using Transformer-based Pretraining. In *International Conference on Artificial Intelligence in Education*, 469–481. Springer.

Taghipour, K.; and Ng, H. T. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, 1882–1891.

Tudor Ionescu, R.; Alexe, B.; Leordeanu, M.; Popescu, M.; Papadopoulos, D. P.; and Ferrari, V. 2016. How hard can it be? Estimating the difficulty of visual search in an image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2157–2166.

Uto, M. 2021. A review of deep-neural automated essay scoring models. *Behaviormetrika*, 48(2): 459–484.

Uto, M.; Xie, Y.; and Ueno, M. 2020. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, 6077–6088.

Valenti, S.; Neri, F.; and Cucchiarelli, A. 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education: Research*, 2(1): 319–330.

Wang, X.; Chen, Y.; and Zhu, W. 2021. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, Y.; Gan, W.; Yang, J.; Wu, W.; and Yan, J. 2019. Dynamic Curriculum Learning for Imbalanced Data Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Wei, Y.; Liang, X.; Chen, Y.; Shen, X.; Cheng, M.-M.; Feng, J.; Zhao, Y.; and Yan, S. 2016. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(11): 2314–2320.

Wu, X.; Dyer, E.; and Neyshabur, B. 2020. When Do Curricula Work? In *International Conference on Learning Representations*.

Xia, L.; Guan, M.; Liu, J.; Cao, X.; and Luo, D. 2020. Attention-Based Bidirectional Long Short-Term Memory Neural Network for Short Answer Scoring. In *International Conference on Machine Learning and Intelligent Communications*, 104–112. Springer.

Zhang, X.; Kumar, G.; Khayrallah, H.; Murray, K.; Gwinnup, J.; Martindale, M. J.; McNamee, P.; Duh, K.; and Carpuat, M. 2018. An empirical exploration of curriculum learning for neural machine translation.