Reducing the Cost: Cross-Prompt Pre-Finetuning for Short Answer Scoring

Hiroaki Funayama[1,2], Yuya Asazuma[1,2], Yuichiroh Matsubayashi[1,2], Tomoya Mizumoto[2] and Kentaro Inui[1,2]

[1] Tohoku University, Sendai, Japan
{h.funa, asazuma.yuya.r7}@dc.tohoku.ac.jp, {y.m, inui}@tohoku.ac.jp,
[2] RIKEN, Tokyo, Japan
tomoya.mizumoto@a.riken.jp

Abstract. Automated Short Answer Scoring (SAS) is the task of automatically scoring a given input to a prompt based on rubrics and reference answers. Although SAS is useful in real-world applications, both rubrics and reference answers differ between prompts, thus requiring a need to acquire new data and train a model for each new prompt. Such requirements are costly, especially for schools and online courses where resources are limited and only a few prompts are used. In this work, we attempt to reduce this cost through a two-phase approach: train a model on existing rubrics and answers with gold score signals and finetune it on a new prompt. Specifically, given that scoring rubrics and reference answers differ for each prompt, we utilize key phrases, or representative expressions that the answer should contain to increase scores, and train a SAS model to learn the relationship between key phrases and answers using already annotated prompts (i.e., cross-prompts). Our experimental results show that finetuning on existing cross-prompt data with key phrases significantly improves scoring accuracy, especially when the training data is limited. Finally, our extensive analysis shows that it is crucial to design the model so that it can learn the task's general property. We publicly release our code and all of the experimental settings for reproducing our results [3].

Keywords: Automated Short Answer Scoring · Natural Language Processing · BERT · domain adaptation · rubrics.

1 Introduction

Automated Short Answer Scoring (SAS) is the task of automatically scoring a given student answer to a prompt based on existing rubrics and reference answers [9,11,16,7]. SAS has been extensively studied as a means to reduce the burden of manually scoring student answers in school education and large-scale examinations or as a technology for augmenting e-learning environments [8,13,19].

---

[3] https://github.com/hiro819/Reducing-the-cost-cross-prompt-prefinetuning-for-SAS.git
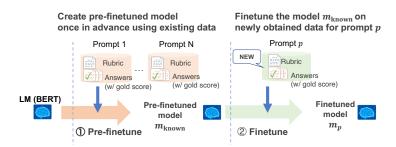
Fig. 1. Overview of our proposed method. We input key phrases, reference expressions, with an answer. We first pre-finetune the SAS model on already annotated prompts and then finetune the model on a prompt to be graded.

However, SAS in the practical application requires one critical issue to be addressed: the cost of preparing training data. Data to train SAS models (i.e. students answers with human-annotated gold score signals) must be prepared for each prompt independently, as the rubrics and reference answers are different for each prompt [2].

In this paper, we address this issue by exploring the potential benefit of using the *cross-prompt* training data, or training data consisting of different prompts, in model training. The cost of preparing training data will be alleviated if a SAS model can leverage cross-prompt data to boost the scoring performance with the same amount of in-prompt data. However, this approach imposes two challenges. First, it is not obvious whether a model can learn from cross-prompt data something useful for scoring answers to a new target prompt, as the new prompt must have different scoring criteria from cross-prompt data available a priori (*cross-prompt generalizability*). Second, in a real-world setting, cross-prompt data (possibly proprietary) may not be accessible when classrooms or e-learning courses train a new model for their new prompts (*data accessibility*). Therefore, we want an approach where one can train a model for a new prompt without accessing cross-prompt data while benefitting from cross-prompt training.

We address both challenges through a new two-phase approach: (i) train (pre-finetune) a model on existing rubrics and answers and (ii) finetune the model for a given new prompt (see Figure 1). This approach resolves the data accessibility issue since the second phase (finetuning on a new prompt) does not require access to the cross-prompt data used in the first phase. Note that the second phase needs access only to the parameters of the pre-finetuned model. On the other hand, it is not obvious whether the approach exhibits cross-prompt generalizability. However, our experimental results indicate that a SAS model can leverage cross-prompt training data to boost the scoring performance if it is designed to learn the task's property shared across different prompts effectively.

Our contributions are as follows. (I) Through our two-phase approach to cross-prompt training, we conduct the first study in SAS literature to alleviate

the need of expensive training data for training a model on every newly given prompt, while resolving the problem of limited accessibility to proprietary cross-prompt data. (II) We conduct experiments on a SAS dataset enriched with a large number of prompts (109), rubrics, and answers and show that a SAS model can benefit from cross-prompt training instances, exhibiting a considerable gain in score prediction accuracy on top of in-prompt training, especially in settings with less in-prompt training data. (III) We conduct an extensive analysis of the model's behavior and find that it is crucial to design the model so that it can learn the task's general property (i.e., a principle of scoring): an answer gets a high score if it contains the information specified by the rubric.

We publicly release our code and all of the experimental settings for reproducing our results at `https://ANONYMIZED`

2 Related work

We position this study as a combination of the use of rubric and domain adoption using cross-prompt data.

To our knowledge, few researchers have focused on using rubrics. A study [18] used key phrases excerpted from rubrics to generate pseudo-justification cues, a span of the answer that indicates the reason for its score, and they showed the model performance is improved by training attention by pseudo-justification cues. [15] proposed a model that utilizes the similarity between the key concept described in rubrics and the answer. Following the utilization of rubrics in previous research, we also use the key phrases listed in the rubric, which are typical expressions used for achieving higher scores.

Domain adaptation in this field is also still unexplored. [17] further pre-trained BERT [4] on a textbook corpus to learn the knowledge of each subject (i.e. science) and report slight improvement. On the other hand, we pre-finetune the BERT on cross-prompt data to adopt the scoring task.

Since SAS has different rubrics and reference answers for each prompt, the use of cross-prompt data still remains an open challenge [6]. As far as we know, the only example is [14]. However, for each new prompt (i.e., target domain in their term), their model was required to be retrained with both in-prompt and cross-prompt data, which leaves the issue of limited accessibility to proprietary cross-prompt data. In contrast, our two-phase approach resolves the data accessibility issue since the second phase (finetuning on a new prompt) does not require access to the cross-prompt data used in the first phase.

3 Preliminaries

3.1 Task definition

Suppose X_p represents a set of all possible student answers for a prompt $p \in P$, and $\mathbf{x} \in X_p$ is an answer. Each prompt has a discrete integer score range $S = \{0, ..., N_p\}$, which is defined in the rubric. The score of each answer is chosen

Prompt

傍線部(3)「それは疑似共生にすぎない」とあるが、筆者がこのように述べるのはなぜか。句読点とも七〇字以内で説明せよ。(*What does the author mean in the phrase "It's only a pseudo symbiosis."? Please answer in 70 words.*)

Analytic criterion A

Rubric
- 「それ」の内容の指摘 (pointing out the content of "it.") - 2pts…

Key phrase
- 緑の庭 (Green garden)
- 緑 (Green)
- 庭 (Garden) ..

Analytic criterion B

Rubric
- 緑の庭は本来の共生のあり方ではないという指摘 (pointing out a green garden is not the original way of symbiosis.) - 3pts…

Key phrase
- 自然と人間の論理のせめぎあいから生まれる本来の共生ではなく(Not the original symbiosis that comes from the struggle between nature and human logic.) …

Analytic criterion C

Rubric
- 疑似共生のあり方の説明(Explanation of the state of pseudo-symbiosis) - 3pts…

Key phrase
- 自然の論理が排除され人間の論理だけで作られたものだから（Because the logic of nature has been eliminated and only the logic of human has been used to create it）…

Student answer

芝生などは人間が考えた論理で、自然の論理を無視して芝生を美しく保つために雑草などをぬいてしまうとそれは人工的な自然
　　　A: 1pts.　　　　　　　　　　　B: 3pts.　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　C: 2pts.
で本物の自然ではないから (If we ignore the logic of nature and remove weeds to keep the lawn beautiful, it is artificial nature and
　　　　　　　　　　　B: 3pts.　　　　　　　　　　　　　　　　　　A: 1pts.　　　　　　　　　C: 2pts.
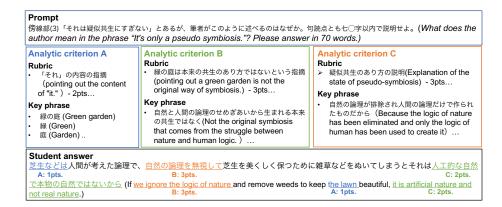not real nature.)

Fig. 2. Example of a prompt, scoring rubric, key phrase and student's answers excerpted from RIKEN dataset [10] and translated from Japanese to English. For space reasons, some parts of the rubrics and key phrase are omitted.

within the range S. Therefore, the SAS task is defined as assigning one of the scores $s \in S$ for each given input $\mathbf{x} \in X_p$.

In this study, we assume that every prompt is associated with a predefined rubric, which stipulates what information an answer must contain to get a score. An answer gets scored high if it contains the required information sufficiently and low if not. Figure 2 shows an example of a prompt with a rubric from the dataset we used in our experiments [10]. As in the figure, the required information stipulated by a rubric may also be presented by a set of key phrases to help human raters and students understand the evaluation criteria. Each key phrase gives an example of wording that gives an answer score high. In the dataset used in our experiments, every rubric provides a set of key phrases, and we utilize such key phrases in cross-prompt training.

In our cross-prompt training setting, we assume that we have some already graded prompts by human raters P_{known} and we then want to grade a new prompt p_{target} automatically. Within this cross-prompt setting, the model is required to score the answers having different score ranges. Therefore, we re-scale the score ranges of all P_{known} and p_{target} to $[0, 1]$, and as a result, the goal of the task is to construct a regression function $m : \bigcup_{p \in P_{\mathrm{known}} \cup \{p_{\mathrm{target}}\}} \{X_p\} \to [0, 1]$ that maps an student answer to a score $s \in [0, 1]$.

3.2   Scoring model

A typical approach to construct a function m is to use deep neural networks. Suppose $\mathcal{D} = ((\mathbf{x}_i, s_i))_{i=1}^{I}$ is training data that consist of the pairs of an actually obtained student answer $\mathbf{x}_i$ and its corresponding human-annotated score s_i. I is the number of training instances. To train the model m, we attempt to minimize

the Mean Squared Error loss on the training data $L_m(\mathcal{D})$:

$$m^* = \operatorname*{argmin}_m \left\{ L_m(\mathcal{D}) \right\}, \quad L_m(\mathcal{D}) = \frac{1}{I} \sum_{(\mathbf{x},s) \in \mathcal{D}} (s - m(\mathbf{x}))^2, \tag{1}$$

where $m(\mathbf{x})$ is a score predicted by the model m for a given input $\mathbf{x}$. Once m^* is obtained, we can predict the score s of a new student answer as: $s = m^*(\mathbf{x})$.

We construct the model m as following. Let $\mathbf{enc}(\cdot)$ as the encoder, we first obtain a hidden vector $\mathbf{h_x} \in \mathbb{R}^H$ from an input answer $\mathbf{x}$ as:

$$\mathbf{h_x} = \mathtt{enc}(\mathbf{x}). \tag{2}$$

Then, we feed the hidden vector $\mathbf{h_x}$ to a linear layer with a sigmoid function to predict a score:

$$m(\mathbf{x}) = \mathtt{sigmoid}(\mathbf{w}^\top \mathbf{h_x} + b), \tag{3}$$

where $\mathbf{w} \in \mathbb{R}^H$ and $b \in \mathbb{R}$ are learnable parameters. In this paper, we used BERT [4], a widely used encoder in various NLP tasks, as the encoder.

4   Method

To leverage cross-prompt training data, we consider the following two-staged training process: (i) We first finetune the model with cross-prompt training instances so that it learns the task's general property (i.e., principles of scoring) shared across prompts, and (ii) we then further finetune the model with in-prompt training instances to obtain the model specific for the target prompt. We call the training in the first stage *pre-finetuning*, following [1]. The questions become what kind of general property can the model learn from cross-prompt training instances and how the model learns.

To address these questions, we first hypothesize that one essential property a SAS model can learn in pre-finetuning is the scoring principle: an answer generally gets a high score if it contains sufficient information specified by the rubric and gets a lower score if it contains less. The principle generally holds across prompts and is expected to be learned from cross-prompt training instances. To learn it, the model needs to have access to the information specified by the rubrics through pre-finetuning and finetuning. We elaborate on this below.

Key phrases As reference expressions for high score answers, we utilize key phrases described in the rubrics as shown in the middle part of Figure 2. Key phrases are representative examples of the expressions that an answer must contain in order to gain scores.

Key phrases are clearly stated in each rubrics. We use those key phrases for each prompt p from the corresponding rubric, and generate a key phrase sequence k_p for p by enumerating multiple key phrases into a single sequence
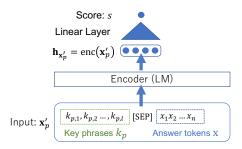
Fig. 3. Overall architecture of our model. We input key phrases and a student answer split by the [SEP] token.

with a comma delimiter. We then use the concatenated sequence $\mathbf{x}'_p$ of tokens k_p, [SEP], and $\mathbf{x}$ in this order, as our model input. For the model without using key phrases, we instead input a prompt ID to distinguish the prompt. We show the overall architecture of the model in Figure 3.

Pre-finetuning We utilize data from already annotated prompts P_{known} to train models for a new prompt p_{target}. For each prompt $p \in P$, there exists a key phrase sequence k_p. We create a concatenated input sequence $\mathbf{x}'_{p,i}$ for the i-th answer of the prompt p as: $\mathbf{x}'_{p,i} = \{k_p, [SEP], \mathbf{x}_{p,i}\}$. Then, we construct data for pre-finetuning as:

$$\mathcal{D}_{\mathrm{known}} = \{(\mathbf{x}'_{p,i}, s_{p,i}) \mid p \in P_{\mathrm{known}}\}_{i=1}^{I}. \tag{4}$$

We pre-finetune the BERT-based regression model on this dataset $\mathcal{D}_{\mathrm{known}}$ and obtain the model m_{known}:

$$m_{\mathrm{known}} = \underset{m}{\mathrm{argmin}} \left\{ L_m(\mathcal{D}_{\mathrm{known}}) \right\}. \tag{5}$$

Next, we further finetune the pre-finetuned model m_{known} on $p \in P_{\mathrm{target}}$ to obtain a model m_p for the prompt p.

$$m_p = \underset{m}{\mathrm{argmin}} \left\{ L_m(\mathcal{D}_p) \right\} \tag{6}$$

5   Experiment

5.1   Dataset

RIKEN dataset We use the RIKEN dataset, a publicly available Japanese SAS dataset[4] provided in [10]. RIKEN dataset offers a large number of rubrics, prompts, and answers ideal for conducting our experiments. As mentioned in

---

[4] https://aip-nlu.gitlab.io/resources/sas-japanese

Section1, we added 10,000 new data annotations (20 prompts with 500 answers each) to the RIKEN dataset.

RIKEN dataset is a collection of annotated high school students' answers for Japanese Reading comprehension questions.[5] Each prompt in the RIKEN dataset has several scoring rubrics (i.e., analytic criterion [10]), and each answer is manually graded based on each analytic criterion independently (i.e., analytic score).

In our experiment, we used 6 prompts (21 analytic criterion), same as [10], from RIKEN dataset as p_{target} to evaluate the effectiveness of pre-finetuning. We split answers for these 6 promts as 200 for train data, 50 for dev set and 250 for test set. For pre-finetuning, we used the remaining 28 prompts (88 analytic criterion), consisting of 480 answers per analytic criterion for training the model and 20 answers per analytic criterion as the dev set.

Following [5], we treat analytic criterion as an individual scoring task since each analytic score is graded based on each analytic criterion independently. For simplicity, we refer to each analytic criterion as a single, independent prompt in the experiments. Thus, we consider a total of 109 analytic criterion as 109 independent prompts in this dataset.

5.2   Setting

As described in Section 3.2, we used pretrained BERT [4] as the encoder for the automatic scoring model and use the vectors of CLS tokens as feature vectors for predicting answers.[6]

Similar to previous studies [10,12,11], we use Quadratic Weighted Kappa (QWK) [3], the de facto standard evaluation metric in SAS, in the evaluation of our models. The scores were normalized to a range from 0 to 1 according to previous studies [12,10]. QWK was measured by re-scaling to the original range when evaluated on the test set. We train a model for 5 epochs in the pre-finetuning process. We then finetune the resulting model for 10 epochs. In the setting without pre-finetuning process, we finetune the model for 30 epochs. These epoch numbers were determined in preliminary experiments with dev set. During the finetuning process, we computed the QWK of the dev set at the end of each epoch and stored the best parameters with the maximum QWK.

To verify the effectiveness of cross-prompt pre-finetuning, we compare the following four settings in experiments; `Baseline`: Only finetune the BERT-based regression model for a target prompt without key phrases, the most straightforward way to construct a BERT-based SAS model. `Key phrase`: Only finetune BERT for a target prompt, we input an answer and key phrases to the model. `Pre-finetune`: Pre-finetune BERT on cross-prompt data, input only an answer. `Pre-finetune & key phrase`: Pre-finetune BERT on cross-prompt data, input an answer and key phrase pairs.

---

[5] Type of question in which the student reads a essay and answers prompts about its content.

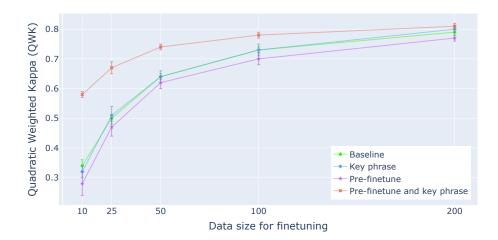[6] We used pretrained BERT models from `https://github.com/cl-tohoku/bert-japanese` for Japanese.

Fig. 4. QWK and standard deviation of four settings described in Section 5.2; `Baseline`, `Key phrase`, `Pre-finetune`, and `Pre-finetune & key phrase`. In the pre-finetuning phase, we use 88 prompts with 480 answers per prompt. We change the amount of data for finetuning as 10, 25, 50, 100, and 200.

5.3   Results

First, to validate the effectiveness of the pre-finetuning with key phrases, we examined the performance of the models for the four settings described in Section 5.2. Here, similar to [10], we experimented with 10, 25, 50, 100, and 200 training instances in the finetuning phase. The results are shown in Fig. 4. We can see that pre-finetune without key phrases slightly lowers the model performance compared to Baseline. As expected, this result indicates that simply pre-finetuning on other prompts is not effective. Similarly, using only key phrases without pre-finetuning does not improve performance. QWK improves significantly only when key phrases are used and when pre-finetune is performed. The gain was notably large when the training data was scarce, with a maximum improvement of about 0.25 in QWK when using 10 answers for finetuning compared to the Baseline. Furthermore, our results indicate that the pre-finetuning with key phrases can reduce the required training data by half while maintaining the same performance. On the other hand, the performance did not improve when we used 200 answers in training, which indicates that pre-finetuning does not benefit when sufficient training data is available. We note that the results of baseline models are comparable to the results of the baseline model shown in [10].

Impact of the number of prompts used for pre-finetuning Next, we examined how changes in the number of prompts affect pre-finetuning: we fixed the total number of answers used for the pre-finetuning at 1,600 and varied the number of prompts between 1, 2, 4, 8, 16, 32, 64. We performed finetuning using
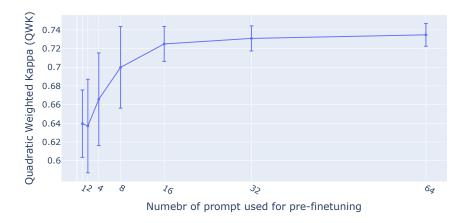
Fig. 5. QWK and standard deviation when the total number of answers used for pre-finetuning is fixed at 1,600 and the number of prompts used is varied from 1, 2, 4, 8, 16, 32, 64. For finetuning, 50 training instances were used.

50 answers for each prompt. The results are shown in Table 5. We see that the performance increases as the number of prompts used for pre-finetuning is increased. This result suggests that the more diverse the answer and key phrases pairs, the better the model understands their relationship. It also suggests that increasing the number of prompts is more effective for pre-finetuning than increasing the number of answers per prompts.

We can see the large standard deviation when the number of prompts used for pre-finetuning is small. We assume that the difference in the sampled prompts caused the large standard deviation, suggesting that some prompts might be effective for pre-finetuning while others are unsuitable for pre-finetuning. The result suggests that a certain number of prompts is needed for training in order to consistently obtain the benefits of cross-prompt learning for each new prompt. We also need some evaluation method for generality of the obtained pre-finetuned model, such as cross-validation among the training prompts.

5.4   Analysis: what does the SAS model learn from pre-finetuning on cross prompt data?

We analyzed the behavior of the model in a zero-shot setting to verify what the model learned from pre-finetuning on cross-prompt data.

First, we examined the performance of the `Pre-finetune & key phrase` model in a zero-shot setting. As a result, we observed higher QWK scores for some prompts, as 0.81 and 079 points in the best-performing two prompts Y14_2-2_2_3-B and Y14_1-2_1_3-D, respectively. The results indicate that the model somewhat learns the scoring principle in our dataset through pre-finetune using

the key phrases; i.e., an answer generally gets a high score if it contains sufficient information specified by the input key phrases.
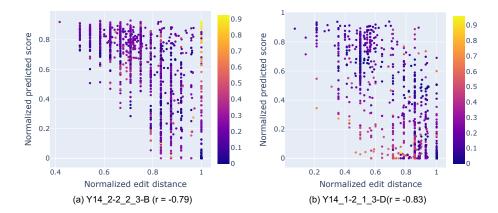

(a) Y14_2-2_2_3-B (r = -0.79)

(b) Y14_1-2_1_3-D(r = -0.83)

Fig. 6. Relationship between (x-axis) the normalized edit distance between the justification cue and key phrases in each answer to (a) Y14_2-2_2_3-B and (b) Y14_1-2_1_3-D and (y-axis) the predicted score in zero-shot settings. The color bars represent the absolute error between a predicted score and a gold score. r indicates the correlation coefficient.

Next, to examine how the key phrases contribute to the scoring, using the above two best-performing prompts, we examined the similarity between the the key phrases and the manually annotated justification cues [10] (substrings of an answer which contributed to gain the score) in the student answer. For the similarity measure, we employ the normalized edit distance. Then we analyze the relationship between the edit distance and the predicted scores by the model.

The results for the two prompts with the highest QWK, Y14_2-2_2_3-B, Y14_1-2_1_3-D, are shown in Figure 6. The color bars represent absolute error between a predicted score and a gold score. The correlation coefficients are -0.79 and -0.83, respectively, indicating a strong negative correlation between edit distance and predicted scores. This suggests that the more superficially distant the key phrases and answer, the lower the predicted model score will be for an answer. We also see that the model correctly predicts a variety of score points for the same edit distance values. We show some examples that have lower prediction error with high edit distance in Table 1. The examples indicate that the model predicts higher scores for answers that contain expressions that are semantically close to key phrases.

Those analysis indicate that the model partially grasp the property of the scoring task, in which an answer gains higher scores if the answer includes an expression semantically closer to the key phrases. Such a feature could contribute to the model's high performance, even when the model could not learn enough answer expression patterns from small training data.

Table 1. Examples of key phrases, answers, predicted scores (Pred.), normalized human annotated scores (Gold.), and normalized edit distance (Dist.). Examples are excerpted from the prompts (1) Y14_2-2_2_3-B and (2) Y14_1-2_1_3-D. Sentences are partially omitted due to the space limitation.

Key phrases	Answers	Pred.	Gold.
(1) 真実よりも幸福を優先する (Prioritize happiness over truth..)	幸福のためにはどうすれば良いか ということについてばかり考える (.. only think about how to realize happiness.)	0.36	0.33
(2) 言葉を尽くして他人を説得する (Convince others with all my words)	説得に努まなければならない.. (..to try hard to convince others.)	0.50	0.50

6 Conclusion

In SAS, answers for each single prompt need to be annotated in order to construct a highly-effective SAS model specifically for that prompt. Such costly annotations are a major obstacle in deploying SAS systems into school education and e-learning courses, where resources are extremely limited. To alleviate this problem, we introduced a two-phase approach: train a model on cross-prompt data and finetune it on a new prompt. Given that scoring rubrics and reference answers are different in every single prompt, we cannot use them directly to train the model. Therefore, we utilized key phrases, or representative expression that answer should contain to gain scores, and pre-finetune the model to learn the relationship between key phrases and answers.

Our experimental results showed that pre-finetuning with key phrases greatly improves the performance of the model, especially when the training data is scarce (0.24 QWK improvement over the baseline for 10 training instances). Our results also showed that pre-finetuning can reduce the amount of required training data by half while maintaining similar performance. As our analysis showed, mere domain adoption by pre-finetuning on cross prompt data is not effective, and it is essential to train the model in terms of the relationship between key phrases and answers to benefit pre-finetuning.

References

1. Aghajanyan, A., Gupta, A., Shrivastava, A., Chen, X., Zettlemoyer, L., Gupta, S.: Muppet: Massive multi-task representations with pre-finetuning. In: EMNLP.

pp. 5799–5811. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.468

2. Burrows, S., Gurevych, I., Stein, B.: The eras and trends of automatic short answer grading. International Journal of Artificial Intelligence in Education **25**(1), 60–117 (2015)

3. Cohen, J.: Weighted Kappa: Nominal Scale Agreement with Provision for Scaled Disagreement or Partial Credit. Psychological bulletin **70**(4), 213–220 (1968)

4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL-HLT. pp. 4171–4186 (Jun 2019). https://doi.org/10.18653/v1/N19-1423

5. Funayama, H., Sato, T., Matsubayashi, Y., Mizumoto, T., Suzuki, J., Inui, K.: Balancing cost and quality: An exploration of human-in-the-loop frameworks for automated short answer scoring. In: AIED. pp. 465–476. Springer International Publishing, Cham (2022)

6. Haller, S., Aldea, A., Seifert, C., Strisciuglio, N.: Survey on automated short answer grading with deep learning: from word embeddings to transformers (2022). https://doi.org/10.48550/ARXIV.2204.03503

7. Krishnamurthy, S., Gayakwad, E., Kailasanathan, N.: Deep learning for short answer scoring. IJRTE **7**, 1712–1715 (03 2019)

8. Kumar, Y., Aggarwal, S., Mahata, D., Shah, R.R., Kumaraguru, P., Zimmermann, R.: Get it scored using autosas — an automated system for scoring short answers. In: AAAI/IAAI/EAAI. AAAI Press (2019). https://doi.org/10.1609/aaai.v33i01.33019662

9. Leacock, C., Chodorow, M.: C-rater: Automated Scoring of Short-Answer Questions. Computers and the Humanities **37**(4), 389–405 (2003), `https://doi.org/10.1023/A:1025779619903`

10. Mizumoto, T., Ouchi, H., Isobe, Y., Reisert, P., Nagata, R., Sekine, S., Inui, K.: Analytic Score Prediction and Justification Identification in Automated Short Answer Scoring. In: BEA. pp. 316–325 (2019). https://doi.org/10.18653/v1/W19-4433

11. Mohler, M., Bunescu, R., Mihalcea, R.: Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments. In: ACL-HLT. pp. 752–762 (2011)

12. Riordan, B., Horbach, A., Cahill, A., Zesch, T., Lee, C.M.: Investigating neural architectures for short answer scoring. In: BEA. pp. 159–168 (2017). https://doi.org/10.18653/v1/W17-5017

13. Roy, S., Dandapat, S., Nagesh, A., Narahari, Y.: Wisdom of students: A consistent automatic short answer grading technique. In: ICON. pp. 178–187. NLP Association of India, Varanasi, India (Dec 2016), `https://aclanthology.org/W16-6324`

14. Saha, S., Dhamecha, T.I., Marvaniya, S., Foltz, P., Sindhgatta, R., Sengupta, B.: Joint multi-domain learning for automatic short answer grading. CoRR **abs/1902.09183** (2019)

15. Sakaguchi, K., Heilman, M., Madnani, N.: Effective feature integration for automated short answer scoring. In: NAACL-HLT. pp. 1049–1054. Association for Computational Linguistics, Denver, Colorado (May–Jun 2015). https://doi.org/10.3115/v1/N15-1111

16. Sultan, M.A., Salazar, C., Sumner, T.: Fast and easy short answer grading with high accuracy. In: NAACL-HLT. pp. 1070–1075. Association for Computational Linguistics, San Diego, California (Jun 2016). https://doi.org/10.18653/v1/N16-1123

17. Sung, C., Dhamecha, T., Saha, S., Ma, T., Reddy, V., Arora, R.: Pre-training BERT on domain resources for short answer grading. In: EMNLP-IJCNLP. pp. 6071–6075. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1628
18. Wang, T., Funayama, H., Ouchi, H., Inui, K.: Data augmentation by rubrics for short answer grading. Journal of Natural Language Processing **28**(1), 183–205 (2021). https://doi.org/10.5715/jnlp.28.183
19. Zhai, X.: Practices and theories: How can machine learning assist in innovative assessment practices in science education. Journal of Science Education and Technology **30**(2), 139–149 (Apr 2021). https://doi.org/10.1007/s10956-021-09901-8