Data Augmentation by Rubrics for Short Answer Grading

Tianqi Wang††, Hiroaki Funayama††, Hiroki Ouchi†† and Kentaro Inui†,††

Short Answer Grading (SAG) is the task of scoring students’ answers for applications such as examinations or e-learning. Most of the existing SAG systems predict scores based only on the answers, and critical evaluation criteria such as rubrics are ignored, which plays a crucial role in evaluating answers in real-world situations. In this paper, we propose a semi-supervised method to train a neural SAG model. We extract keyphrases that are highly related to answers scores from rubrics. Weights to words of answers are calculated as attention labels instead of manually annotated attention labels, based on span-wise alignments between answers and keyphrases. Only answers with highly weighed words are used as attention supervision. We evaluate the proposed model on two analytical assessment tasks of analytic score prediction and justification identification. Analytic score prediction is the task of predicting the score of a given answer for a prompt, and Justification identification involves identifying a justification cue in a given student answer for each analytic score. Our experimental results demonstrate that both performance of grading and justification identification is improved by integrating attention semi-supervised training, especially in a low-resource setting.

Key Words: SAG, Rubrics, Semi-Supervision, Short Answer Grading, Short Answer Scoring, Data Augmentation

1 Introduction

In a pedagogical setting, assignments (e.g., examinations, homework, etc.) are crucial for assessing students’ knowledge. Students are generally required to answer questions in relatively short texts. Educators can have a clear idea of how well the students are learning from their answers. Simultaneously, such assignments allow students to detect and correct their errors and misunderstandings (Sychev et al. 2020). Figure 1 shows a typical example where students are asked to explain a sentence that appears in an essay to test their reading comprehension.

For grading assignments, each question, henceforth prompt, is provided along with scoring rubrics. Educators typically assess student answers following the scoring rubrics. Figure 2 gives an example. The rubrics explain in detail the rules to grade each item. Student answers are
**Question:**

A typical example of a prompt to a provided essay to test the reading comprehension of students. An example of as student's answer is also shown.

**Student answer:**

Reference answer:

Rubric:

Fig. 2 An example of rubrics explaining the rules for scoring. There are four items labeled from A to D to score. Key elements are defined by rubrics with some keyphrases (in quote marks) for each item, and answers get points if it contains any key element. Answers are scored by each item, and the final score is calculated as the sum of item scores.
scored on four different criteria, labeled from A to D in the reference answer. The holistic score of an answer is assumed to be the sum of all the item scores.

One of the crucial ingredients of rubrics is the key elements it provides. **Key elements** are concepts or information defined by the rubrics that an answer needs to contain to receive points. Rubrics provide some keywords or keyphrases as examples of the key elements. For instance, according to the rubrics shown in Figure 2, the phrase *(in the western world)* is a keyphrase to define a key element for item A, thus answers will receive 2 points if they contain a similar phrase. At the same time, rubrics provide some examples of incorrect keyphrases that a student answer may contain. (e.g., 0 points for the phrase *(foreign countries)* for item A).

Due to the diversity of open-ended questions in answer scoring, educators’ workload can be excruciatingly time-consuming, especially for online courses with a large number of students. The task of Short Answer Grading (SAG) has been proposed to assist educators with grading. SAG is the task of estimating scores of short-text answers written as an answer to a given prompt, on the basis of whether the answer satisfies the rubrics prepared by a human in advance (Mohler et al. 2011; Funayama et al. 2020; Mizumoto et al. 2019). The SAG systems play a central role in providing stable and sustainable scoring in repeated and large-scale examinations and (online) self-study learning support systems (Attali and Burstein 2006; Mizumoto et al. 2019; Shermis et al. 2010; Leacock and Chodorow 2003; Burrows et al. 2015). Two analytical assessment tasks of SAG are formalized in (Mizumoto et al. 2019): i) analytic score prediction and ii) justification identification. Analytic score prediction gives scores to answers, and justification identification identifies answer segments that contribute to the scoring, showing the reasons for the corresponding prediction of analytic scores. We discuss the details of task settings in Section 2.

SAG has been studied mainly with machine learning-based approaches. The task is considered as inducing a regression model from a given set of manually scored sample answers (e.g., training instances). As observed in a variety of other NLP tasks, recently proposed neural models have been yielding strong results (Riordan et al. 2017; Mizumoto et al. 2019).

SAG is generally a low-resource task, that the scarcity of training data is an issue for SAG tasks. (Heilman and Madnani 2015) investigate the importance of the size of training data on non-neural SAG models with discrete features. Recently proposed neural models for SAG are trained on each prompt independently (Riordan et al. 2017; Mizumoto et al. 2019). Thus, the lack of data is a crucial issue for neural models as well. For the two assessment tasks we introduced earlier, annotation of both scores and justification cues are needed to train the neural models. The workload of annotation can be hefty, especially for the annotation of justification cues. (Mizumoto et al. 2019) developed a dataset with annotated analytic scores and an attention supervised model,
and the experimental results show that the performance is improved by attention supervision. However, the model is trained on each prompt, thus it is necessary to prepare annotated data as attention supervision for each prompt, leading to a high cost even in low resource settings. To solve this issue, we introduce rubrics to SAG model instead of using annotated justification cues, in order to reduce the cost of the annotation of justification cues.

This paper is the first study that explores how to incorporate rubric information into neural SAG models in order to reduce the required amount of training data. We extract keyphrases that are highly related to the scores of answers from rubrics, and calculate the weights of the answers’ words based on span-wise alignments between answers and the extracted keyphrases. We then propose a semi-supervised method to train a neural SAG model. Weights to highly weighed words are used as attention supervisory signals for semi-supervised learning, instead of manually annotated data.

Our experimental results demonstrate that the performance of SAG is improved by the training data augmented with pseudo attention supervision, especially in a low-resource setting.

2 Task and Evaluation

We evaluate the proposal model on two analytical assessment tasks of SAG formalized in (Mizumoto et al. 2019): i) analytic score prediction and ii) justification identification. Analytic score prediction is the task of predicting the score of a given answer for a prompt. Given a number of answer texts such as those shown in Figure 1, an SAG model is expected to output either scores (Mohler et al. 2011) or labels (e.g. correct, natural, incorrect, etc.) (Dzikovska et al. 2013). Given a student answer that consists of n tokens, \( t^{ans}_1: t^{ans}_n \), the goal is to predict the analytic score \( s \in \mathbb{R} \).

As shown in Figure 2, an answer gains analytic scores for each item (e.g. (A)∼(D)) following corresponding analytic criterion (Try one’s best for 3 points, e.g.). In this paper, we train the model on each item independently to predict the analytic scores. To evaluate the precision of analytic score prediction, we use quadratic weighted kappa (QWK) (Cohen 1968), which is commonly used in the SAG literature.

Justification identification involves identifying a justification cue in a given student answer for each item as interpretation to the prediction of the analytic score. A justification cue is a segment of an answer that contributes to the analytic score. An example of justification cues for item D is shown in Figure 3. As the manually annotated justification cue, the phrase 言葉を尽くして (try their best) in the answer refers to 言葉を尽くして (try one’s best) in the rubric, and 自
Fig. 3 An example of justification cues in a student answer for item D. There are seven keyphrases defined in the rubric. Tokens in green are manually annotated justification cues. Tokens in yellow are justification cues generated by answers and key elements. A deeper color represents higher relatedness of the justification cue to the key elements.

分の考えを伝える (convey their ideas) refers to (自分の考えを) 伝えようとする (convey one’s thoughts). Formally, given a student answer $t_{ans}^{1:n} = (t_{ans}^1, t_{ans}^2, ..., t_{ans}^n)$, the goal is to identify the segment $p_{a:b} = (t_{ans}^{a}, t_{ans}^{a+1}, ..., t_{ans}^{b})$, where $1 \leq a \leq b \leq n$.

In this paper, to evaluate performance on justification identification, we predict justification cues depending on the predicted attention, and calculate F1 score following the method in (Mizumoto et al. 2019) for evaluation.

3 Related Work

Many existing SAG studies have mainly focused on exploring better representations of answers and similarity measures between student answers and reference answers. A wide variety of methods have been explored so far, ranging from Latent Semantic Analysis (LSA) (Mohler et al. 2011), edit distance-based similarity, and knowledge-based similarity using WordNet (Pedersen et al. 2004; Magooda et al. 2016) to word embedding-based similarity (Sultan et al. 2016). It is reported in (Riordan et al. 2017) that neural network-based feature representation learning (Taghipour and Ng 2016) is effective for SAG.

In contrast to the popularity of learning answer representations, the use of rubric information
for SAG has been received little attention so far. In (Sakaguchi et al. 2015), the authors compute similarities, such as BLEU (Papineni et al. 2002), between an answer and each key element in a rubric, and use them as features in a support vector regression (SVR) model. Text patterns are generated from reference answers and rubrics in (Ramachandran et al. 2015), and the results demonstrate that the automatically generated pattern outperforms the manually generated regexp pattern.

Simultaneously, data augmentation techniques have been developed to achieve better performance with a limited dataset. (Mintz et al. 2009) assume that if two entities participate in a relation, all sentences that mention these two entities express that relation. The assumption is relaxed by (Riedel et al. 2010) as expressed-at-least-once assumption, saying that if two entities participate in a relation, at least one sentence that mentions these two entities might express that relation. To augment the dataset for the relation classification task, (Ye and Ling 2019) found all sentences containing those entities for each pair of entities that appear in some Freebase relation in a large unlabeled corpus extract textual features to train the classifier. (Wei and Zou 2019) proposed an Easy Data Augmentation (EDA) method. EDA consists of four operations: synonym replacement, random insertion, random swap, and random deletion.

3.1 Neural baseline model

We take the simple but effective neural model (Taghipour and Ng 2016; Riordan et al. 2017) as the baseline of the neural model in this paper, as is shown in Figure 4(a). Given an answer of \( n \) tokens, \( t_{1:n}^{ans} = (t_1^{ans}, t_2^{ans}, \ldots, t_n^{ans}) \), the embedding layer outputs an embedding vector \( t_{i}^{ans} \in \mathbb{R}^{d_{token}} \) for each token \( t_i^{ans} \in t_{1:n}^{ans} \). Taking a sequence of these vectors \( t_{1:n}^{ans} = (t_1^{ans}, t_2^{ans}, \ldots, t_n^{ans}) \) as input, the Bi-LSTM layer then produces a contextualized vector \( h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] \) by concatenating the forward hidden state of the Bi-LSTM \( \overrightarrow{h_i} \in \mathbb{R}^{d_{lstm}} \) and the backward one \( \overleftarrow{h_i} \in \mathbb{R}^{d_{lstm}} \). The attention layer then assigns an attentional weight \( \alpha_i \) to each contextualized vector \( h_i \) as follows:

\[
\alpha_i = \frac{\exp(e_i)}{\sum_{k=1}^{n} \exp(e_k)},
\]

where \( W \in \mathbb{R}^{d_{lstm} \times d} \) is trainable parameter matrix and \( m \in \mathbb{R}^{d} \) is a trainable parameter vector. Then the regression layer computes an analytic score \( s \) by the weighted sum:

\[
f = \sum_{i=1}^{n} \alpha_i h_i,
\]

\[
s = \text{sigmoid}(w^\top f + b),
\]

188
where $w \in \mathbb{R}^d$ is a trainable parameter vector and $b \in \mathbb{R}$ is a trainable bias parameter. Owing to the sigmoid function, each analytic score takes a value from 0 to 1. Analytic scores are rescaled back to the original score range before evaluation on QWK.

The attention mechanism (Equation 1) is used not only for analytic score prediction but also for justification identification (Mizumoto et al. 2019). Specifically, based on the attention values $\alpha_i$ for each token $t_i^{ans}$ in Equation 1, we extract a set of token indices as justification cues $C$:

$$\alpha_{\text{max}} = \max_{i=1, \ldots, n} \alpha_i,$$

$$C = \{i \in [1, n] | \alpha_{\text{max}} - \alpha_i < \gamma\},$$

where $\gamma$ is a hyperparameter selected by using the development set.

To train the model, the mean squared error (MSE) is used as the loss function to minimize:

$$l = \frac{1}{N} \sum_{j=1}^{N} (s_j^{\text{gold}} - s_j)^2$$

189
where $s_j^{\text{gold}}$ and $s_j$ are the gold and predicted scores for the $j$-th answer, respectively. $N$ is the number of the answers in the training set.

For justification cue identification, (Mizumoto et al. 2019) provides a dataset with manual annotations for ground truth justification cues, and the model can be trained by using them as attention supervision, as shown in Figure 4b. The loss is calculated as follows:

$$l = \frac{1}{N} \sum_{j=1}^{N} (s_j^{\text{gold}} - s_j)^2 + \frac{1}{Nn} \sum_{j=1}^{N} \sum_{i=1}^{n} (\alpha_{j,i}^{\text{gold}} - \alpha_{j,i})^2$$

where $\alpha_{j,i}^{\text{gold}}$ and $\alpha_{j,i}$ are the manually annotated attention and predicted attention for the $i$-th token in the $j$-th answer, and $n$ is the number of tokens in each answer. We take this attention supervised method as a reference for comparison on performance in this paper.

In (Mizumoto et al. 2019), the SAG model is designed to compute the sum of analytic scores for each item as the holistic score for a given answer. The performance of those models is evaluated in terms of the accuracy of predicted holistic scores. In this study, however, we focus on the performance of predicting item-wise analytic scores (rather than holistic scores) to closely analyze the effects of using rubrics, which are provided for each item. We, therefore, train models and evaluate their performance for each item separately.

4 Proposed Method

4.1 Key idea

(Mizumoto et al. 2019) reported an improvement in the performance of analytic score prediction from an attention supervised model. The annotations for ground-truth justification cues $\alpha^{\text{gold}}$ are used as supervisory signals to learn the attention mechanism. For example, in Figure 4 (b), we have the supervisory signals for attention simply based on the annotated justification cues. Attention to tokens of justification cues (try their best to convey their ideas) is set to 1, and attention to other tokens is set to 0.

Instead of manual annotation of justification cues, we consider using the rubrics to automatically identify which span of a given answer should be attended. If this identification can be carried out reasonably accurately, the identified cues can be used as pseudo supervisory signals of attention to train the attention mechanism.

The idea is illustrated in Figure 4 (c). For each input answer $t^{\text{ans}}$ from a training set, we first search for the justification cues by matching every candidate span in $t^{\text{ans}}$ with a keyphrase $t^k$ from the rubric. In the figure, the yellow shades depict the weights of spans of the identified...
justification cues. If a span that is likely to be a justification cue is identified, we then use it as a pseudo supervisory signal to train the attention prediction component. Of course, such a span may not be identified because, for example, (i) the list of keyphrases in the rubric may not exhaust all the potential variety of justification cues in answers, or (ii) a given answer may not contain any contributive keyphrase. In such a case, we do not use that instance for supervising attention prediction but use only its gold score to train the overall model as in the attention-unsupervised baseline model (Figure 4 (a)).

Notice that the rubrics also provide some zero points keyphrases such as 「外国」は不可 (‘Foreigner’ is invalid). Even though those zero points phrases do not contribute any points to an answer, we consider them as cues for mistakes that are easily made by students, and feed into the SAG model as part of supervision to train the model to be able to distinguish mistakes. However, because very little training data includes zero points keyphrases, especially in low resource settings where the size of training data is limited, no difference was observed by introducing zero points keyphrases during experiments.

4.2 Data augmentation based on span-wise matching

A more concrete example for data augmentation is given in Figure 5. First, answers and rubrics are tokenized by MeCab (Kudo 2006), and keyphrases, such as 「他人を」説得する,
are extracted from the given rubrics. We then search for justification cues by matching a given answer to the keyphrases. Weights of answer tokens are calculated based on span-wise matching between answers and keyphrases as illustrated in the figure.

Given an answer of \( n \) tokens \( t_{\text{ans}}^i \), we first enumerate all possible spans: \( P = \{a:b|1 \leq a \leq b \leq n\} \), where each span \( p_{a:b} = (t_{\text{ans}}^a, \ldots, t_{\text{ans}}^b) \) is a subsequence of the tokens in the answer. Each keyphrase is a sequence of \( m \) tokens: \( t_k^1: m \). The similarity score \( w_{a:b} \) between a keyphrase \( t_k^1:m \) and each span \( p_{a:b} \) is calculated by some similarity (distance) measure (e.g., Levenshtein distance): \( w_{a:b} = \text{sim}(t_k^1:m, p_{a:b}) \). Based on the similarities, we define the matching score \( w_k \) to indicate how well the keyphrase \( t_k^1:m \) matches (or is entailed in) the answer. Specifically, we define \( w_k \) as the highest similarity score between the keyphrase and each span \( p_{a:b} \):

\[
w_k = \max_{p_{a:b} \in P} w_{a:b}.
\]

Here, based on the pseudo justification cues (spans) \( p^k \), we define a pseudo supervisory signal \( \alpha_i^{\text{pseudo}} \) of each token \( t_{\text{ans}}^i \) in an answer \( t_{\text{ans}}^i \) for learning the attention mechanism, instead of the ground-truth \( \alpha_i^{\text{gold}} \) (Equation 2). Specifically, if a token \( t_{\text{ans}}^i \) is contained in a span \( p^k = (t_k^1, \ldots, t_k^K) \), we use the similarity score \( w_k \) as the pseudo supervisory signal:

\[
\alpha_i^{\text{pseudo}} = \max_{w_k \in W_i} w_k,
\]

In the former equation, we judge whether a token \( t_{\text{ans}}^i \) is contained in a span \( p^k \), which is extracted by using each keyphrase \( k \in (k_1, \ldots, k_K) \), and if so, we extract the similarity score \( w_k \) associated with the span \( p^k \). A token may be contained in multiple spans, thus we pick the highest one of the similarities \( w_k \in W_i \) in the latter equation.

### 4.3 Semi-supervised learning with pseudo labels

We propose an attention semi-supervised method for SAG. Only attention to justification cues similar to keyphrases will be used as pseudo signals for attention. Given an answer \( a \), if the maximum value of attention to justification cues is not high enough, the answer will not be used for attention-supervised learning. A binary label \( \eta \) is attached to each answer \( a \) to prepare training data for semi-supervised learning,

\[
\eta_j = \begin{cases} 
1, & \max(\alpha_j^{\text{pseudo}}) \geq T \\
0, & \max(\alpha_j^{\text{pseudo}}) < T 
\end{cases}
\]
where $\alpha_{j}^{\text{pseudo}} = (\alpha_{j,1}, \cdots, \alpha_{j,n})$ is a set of the pseudo attention signals for the tokens $(t_{j,1}^{\text{ans}}, \cdots, t_{j,n}^{\text{ans}})$ in an answer $a_j$ generated by Equation 3 and threshold $T$ is a hyperparameter. We consider justification cues are matched in answers where $\eta = 1$.

The loss function is designed as follows:

$$l = \frac{1}{N} \sum_{j=1}^{N} (s_{j}^{\text{gold}} - s_{j})^2 + \frac{1}{Nh} \sum_{j=1}^{N} \sum_{i=1}^{n} (\eta_{j}\alpha_{j,i}^{\text{pseudo}} - \eta_{j}\alpha_{j,i})^2$$

where $N$ is the number of answers, and $n$ is the number of tokens in each answers.

Note that answer $a_j$ is considered containing no justification cues if $\eta_j = 0$, and the corresponding loss of attention is always 0. In this way, only answers with a significant match to keyphrases are used for attention-supervised learning.

5 Experiments

Our research question is how much benefit we can gain from the proposed method by introducing rubrics to SAG models. There are three targets we want to explore through experiments to answer this question: i) what is the best settings of proposed attention semi-supervised training, ii) what is the best way to import rubric to SAG models, and iii) how much benefit we can obtain from the proposed semi-supervised method. To address these questions, we first run the proposed model in a low-resource setting with various hyperparameters to find the best settings. Then we propose four simple rubric-based models and compare the performance with our model. Finally, we compare the proposed model with the attention unsupervised baseline model to show the improvement.

5.1 Settings

We applied our proposed method on the dataset provided by (Mizumoto et al. 2019) \(^1\). In total, the dataset includes 6 prompts ($Q1 \sim Q6$), with 1600 answers as training data, 250 answers as development data, and 250 answers as test data for each prompt. Similar to the example shown in Figure 2, answers are scored by multiple items, and justification cues are manually annotated for each answer, where the corresponding tokens are annotated to 1 while the other tokens are annotated to 0. Three of the prompts ($Q1$, $Q2$, $Q3$) each contain 4 items, and the remaining prompts ($Q4$, $Q5$, $Q6$) each contain three items, thus equating to 21 items to grade.

We pre-train 100-dimensional Word2Vec word embeddings (Mikolov et al. 2013) on Japanese

\(^1\) https://aip-nlu.gitlab.io/projects/sas-j
Wikipedia data to initialize the word embedding layer. Little gain is obtained from fine-tuning the embedding layer according to (Riordan et al. 2017); thus we freeze the embedding layer during the training step in our experiments to reduce the number of parameters to learn in low-resource settings. The Bi-LSTM layer’s dimension is set to $d_{\text{lstm}} = 250$, and the dropout probability is set to 0.5. The parameter $d$ for $W$ and $m$ of the attention layer is set to 100. $W$ and $m$ are initialized randomly from normal distribution, with the standard deviation set to 0.01.

We train our model on each item, optimized with the Adam (Kingma and Ba 2014) optimizer, and the learning rate is set to 0.001. Experiments are repeated for 5 times with different random seed from 0 to 4 for initialization, and the average results across all random seeds are used as the final results. To explore the performance in low-resource settings, we train our model on various sizes of training data, ranging from 48 to 1600 (3%, 6%, 12.5%, 25%, 50%, 100%).

5.2 Rubric-based baselines

To explore different ways to introduce rubric information to the SAG model, we proposed three simple baseline models that score answers with rubric information, and apply regular expression (regexp) matching instead of span-wise matching to our proposed method for comparison.

Rule-based model on top of regexp matching (RGEXP-RULE): As described in Figure 2, an answer is scored by steps (D(1) and D(2) for item D). On each step, corresponding points are assigned to the answer if the answer contains any required keywords, and the sum of scores assigned in each step is output as the final score for the answer. Following this rule, we first create regular expressions for keywords based on the rubrics. For example, regex for ‘哲学 (の営み) は’ is constructed as ‘哲学 (の営み)?は’. Then we identify keywords in the given answer by regexp matching for each step, and output the sum of corresponding scores of matched keywords in each step as the prediction of answer score. Only the keywords with the highest score will be selected if multiple keywords are matched in any step.

Rule-based model on top of span-wise matching (SPAN-RULE): This method works similarly to RGEXP-RULE introduced above, but a span-wise matching method based on edit-distance is used instead of regexp matching. The details of the span-wise matching method are introduced in Section 6. Note that there is a hyperparameter $T$ for span-wise matching. We apply different values of $T$ from 0.1 to 1.0, and take the value leading to the highest mean QWK over all prompts.

Neural model with rubrics (CONCAT-RGRS): As demonstrated in Figure 6, we introduce rubric information to the neural model. A number of keyphrases are extracted from rubrics. We encode the answer and key elements by an encoder consisting of a word embedding layer,
a BiLSTM layer and an attention layer, with settings introduced in Section 5.1. The encoded answer $a$ and encoded keyphrases $k_1, k_2, \ldots, k_K$ are concatenated to $f$, and feed into a regression layer to predict the score of the given answer.

Our method with regexp matching (RGEXP-RGRS): Instead of proposed span-wise matching based on edit distance, we apply regexp matching between answers and keyphrases to generate pseudo attention supervisory signals. Then the model is trained with the augmented training data, and with the same settings as introduced in Section 5.1.

The results are discussed in Section 6.3.

6 Results and Discussion

6.1 Settings of proposed model

We generate pseudo supervisory signals for attention semi-supervised training following the method introduced in Section 4. To simplify the process, we calculate the weights of span $p_{a:b}$ based on Levenshtein edit distance (Levenshtein 1966) between $p_{a:b}$ and keyphrase $t_{1:m}^k$:

$$w_{a:b} = 1 - \frac{\text{edit\_distance}(p_{a:b}, t_{1:m}^k)}{\max(b - a + 1, m)}, 0 \leq w_{a:b} \leq 1$$

We limit the length of spans to $0 \leq b - a < 3$ to avoid lengthy spans, and set $w_{a:b}$ to 0 if $w_{a:b} < T$ to avoid influence from low-weighted tokens, as is shown in Figure 7.

The size of training data as attention supervision is controlled by a threshold $T$. With a low value of $T$, spans will be identified as justification cues even if the similarity to keyphrases is low, and most of the training data will be used as pseudo attention supervision. On the other side, with a higher value of $T$, very limited training data containing justification cues that are similar enough to keyphrases will be used as attention supervision.
Fig. 7 Influence of $T$ to pseudo supervisory signal on attention. Tokens marked by a deeper color are closer to key elements provided in rubrics. With greater values of $T$, only tokens similar enough to keyphrases are used as supervisory signals.

We compute MSE between predicted attention and attention supervision as the loss of attention as we introduced in Section 4.3, thus we binarize labeled attention $\alpha_{\text{pseudo}}$ to $\beta_{\text{pseudo}}$ in order to enhance the effectiveness of attention supervision.

$$
\beta_{j,i} = \begin{cases} 
1, & \alpha_{j,i} > T \\
0, & \text{otherwise}
\end{cases}
$$

To demonstrate the influence of value $T$, we manually set $T$ to 0, 0.25, 0.5 and 0.9. Figure 8 shows statistical information on answers containing justification cues. The blue bars indicate the size of training data containing justification cues ($\eta = 1$), and the red polylines represent the MSE between $\beta_{\text{pseudo}}$ and manually annotated gold attention. Note that MSE tends to be higher for lower $T$ values because spans unrelated to key elements are considered as justification cues.

The performance of the proposed model in low-resource settings (3% training data) with various values of $T$ is shown in Figure 9. Both performance with $\alpha_{\text{pseudo}}$ and $\beta_{\text{pseudo}}$ are shown for comparison. The blue bars on left represent performance with $\alpha_{\text{pseudo}}$, the red bars on right represent performance with $\beta_{\text{pseudo}}$. We also show the performance of the ATT-UNSUP model and ATT-SUP model for reference.

The results demonstrate that the neural SAG model benefits from attention semi-supervised training. Both QWK for analytic score prediction and F1 score for justification identification are improved compared to the attention unsupervised baseline model, even though the pseudo
Fig. 8  Statistics of training data as attention supervision generated with various value of threshold $T$. The size of data is indicated by the blue bars. The red polylines show MSE of pseudo supervisory signals to manually annotated attention.

Fig. 9  Performance of the proposed model in a low-resource setting, where training size is set to 3%, with various values of threshold parameter $T$ from 0.0 to 0.9. The blue bars on left represent performance with generated pseudo attention ($\alpha^{\text{pseudo}}$), and the red bars on right represent performance with the binary pseudo attention ($\beta^{\text{pseudo}}$). The performance of the ATT-UNSUP model and ATT-SUP model are for comparison.

supervisory signal for attention is not necessarily correct compared to manually annotated data.

The results also demonstrate that the performance of the proposed model changes with the values of $T$, and performance with binary attention is more sensitive to $T$. With a proper value of $T$, the binary pseudo supervisory signal for attention leads to better improvement in performance. We select the value of $T$ for each prompt by following steps. We run the proposed model in low-resource settings (3% training data) with $\beta^{\text{pseudo}}$ and repeat the experiments with various values...
of $T$. Values of $T$ leading to the best QWK on the development data for each prompt are selected, listed in Table 1, and will be used as the hyperparameter for other scales of training data.

We discuss the performance of proposed model in the following section.

6.2 Performance of proposed model

We compare the proposed method to the attention unsupervised baseline model (ATT-UNSUP). Because the attention supervised model (ATT-SUP) is trained with manually annotated justification cues, we show the performance for reference. The difference in training data between the models is listed in Table 2, and the mean results over all of the prompts are shown in Figure 10. The figure indicates that the baseline models achieved comparable performance to the reports from (Mizumoto et al. 2019), and the attention supervision improves performance on score prediction, especially in low-resource settings. Performance on justification identification is also improved over all the training sizes.

Compared to the ATT-UNSUP model, the performance is improved by semi-supervised learning with pseudo supervisory signals for attention, especially in low-resource settings. It is also important to mention that we achieved comparable performance to the baseline with a larger size of training data, indicating that our proposed semi-supervised method does not harm the performance when a large amount of training data is available. At the same time, F1 for justification identification is also improved through various training sizes, as shown in Figure 10(b), indicating better interpretability of our proposed method on the scoring results.

Still, ATT-SUP performs best overall, especially on justification identification. One of the reason is the justification cues used as pseudo supervisory signals are identified by a simple span-wise

| Prompt     | Q1/A | Q1/B | Q1/C | Q1/D | Q2/A | Q2/B | Q2/C | Q2/D | Q3/A | Q3/B | Q3/C |
|------------|------|------|------|------|------|------|------|------|------|------|------|
| Threshold  | 0.5  | 0.1  | 0.1  | 0.5  | 0.5  | 0.0  | 0.3  | 0.5  | 0.3  | 0.1  | 0.4  |
| Prompt     | Q3/D | Q4/A | Q4/B | Q4/C | Q5/A | Q5/B | Q5/C | Q6/A | Q6/B | Q6/C |
| Threshold  | 0.2  | 0.5  | 0.9  | 0.5  | 0.6  | 0.5  | 0.6  | 0.0  | 0.4  | 0.6  |

**Table 1** Best threshold $T$ selected based on development data.

|                       | ATT-UNSUP(base) | SPAN-RGRS(proposal) | ATT-SUP(reference) |
|-----------------------|-----------------|----------------------|--------------------|
| Justification identification | unsupervised | Augmented data | Gold data |
| Score prediction      | Gold data      | Gold data          | Gold data          |

**Table 2** Training data of each model.
Wang et al. Data Augmentation by Rubrics for Short Answer Grading

matching method based on edit-distance. However, SPAN-RGRS model achieves a comparable performance to ATT-SUP, without any requirement on manually annotation of justification cues.

Performance trained with 3% training data on each prompt is listed in Table 3. ATT-UNSUP represents the performance of the attention unsupervised baseline model, and SPAN-RGRS represents the performance of our proposed model. The table shows that the proposed model outperforms the ATT-UNSUP model for almost all of the prompts. Note that the performance of both the baseline model and the proposed method vary with items because of the diversity of

![Graph showing performance improvement](image)

(a) Performance of score prediction  
(b) Performance of justification identification

**Table 3 Performance in low-resource settings.** ATT-UNSUP is the attention unsupervised baseline model, and SPAN-RGRS is our proposed model.
prompts and answers. The benefit we obtain from the proposed method also varies with items for the same reason.

For instance, the rubric for prompt Q1/A is simple that answers containing 西洋 or ヨーロッパ get 2 points. Compared to other prompts, more benefit is obtained from our proposed method, and the performance is close to the attention supervised baseline model. Two reasons are leading to these results. First, because the keyphrases included by the rubrics are simple and the number of keyphrases is limited, the scoring rules are easy to learn. Second, for the same reason, the generated pseudo attention signals are more reliable than other prompts. The rubric for prompt Q1/D is more complicated, as is shown in Figure 2. The answers are graded by two steps, and keyphrases are longer and more diverse. Hence the improvement from our proposed method is limited.

6.3 Comparison against rubric-based baselines

We compare the performance to rubric-based baseline models introduced in Section 5.2, and the mean performances over all prompts in low resource settings (3% training data) is listed in Table 4. The performance on each prompt is shown in Figure 11. The prompts are sorted by the performance of our proposed SPAN-RGRS model. The attention unsupervised neural baseline model ATT-UNSUP is included for reference.

Firstly, note that CONCAT-RGRS performs close to ATT-UNSUP. It demonstrates that it is vital to find a proper way to introduce rubric information to SAG models. Directly encoding rubrics into features does not help improve the performance. Secondly, SPAN-RULE performs much better than RGEXP-RULE, indicating that the proposed span-wise matching method works better than regexp matching. Similarly, SPAN-RGRS performs better than RGEXP-RGRS for

| Import Rubric | Justification identification | Scoring | QWK  |
|----------------|-------------------------------|---------|------|
| SPAN-RULE      | span-wise matching            | rule based | rule based | 0.510 |
| RGEXP-RULE     | regexp matching               | rule based | rule based | 0.353 |
| RGEXP-RGRS     | regexp matching               | learn from augmented data | regression | 0.666 |
| SPAN-RGRS      | span-wise matching            | learn from augmented data | regression | 0.697 |
| CONCAT-RGRS    | concat                        | unsupervised | regression | 0.653 |
| ATT-UNSUP      | —                             | unsupervised | regression | 0.659 |

Table 4 Performance of rubric-based models in low resource settings (3%, 48 instances)
the same reason. Note that compared to RGEXP-RGRS, SPAN-RGRS model does not need manually designed regexp for matching, but still achieves a better result compared to RGEXP-RGRS. Furthermore, Figure 11 shows that both RGEXP-RULE and SPAN-RULE achieve outstanding performance for some specific cases (Q1/A, e.g.), where the rubrics are simple, and keyphrases are easy to match to answers. However, for prompts with complicated rubrics, keyphrases are more challenging to extract, and answers are more diverse to match keyphrases, leading to a very low QWK. As a typical example, the SPAN-RGRS model works well on Q6/B, but the REGEXP-RULE model achieves a very deficient performance. One of the rubrics for Q6/B is: 

(The answer is acceptable if “the truth is disregard compared to happiness” can be entailed). Because of the open rubric, answers to Q6/B are more diverse than other prompts. As a result, it is more difficult to identify correct justification cues by regular expression matching. Thus most of the answers are scored to 0 by REGEXP-RULE model. On the other side, the proposed span-wise matching method is more tolerant of justification cues. Moreover, because we train an attention layer and a regression layer for the REGEXP-RGRS model to predict scores, with not only justification cues but also answer score pairs, the model gives relatively acceptable scoring results, even for the cases when it fails to identify the correct justification cues. Hence, neural models are more stable than rule-based models.
7 Conclusion

Annotation of scores and justification cues are needed to train neural models for SAG. Particularly, the annotation on justification cues brings a heavy workload to human annotators. Considering that attention supervision is required for each prompt, the preparation of training data is still time-consuming, even for low-source settings.

To reduce the workload on annotation, we proposed a method to augment training data with rubrics, and generate pseudo attention supervision instead of manually annotated data. We identify justification cues by matching answers to keyphrases provided by rubrics. Pseudo supervisory signals for attention are created based on matching scores, and only justification cues that are significantly matched are used for attention supervision.

The proposed method improves the performance of SAG on both analytic score prediction and justification identification tasks, compared to the attention unsupervised baseline model, especially in low-resource settings. The performance is comparable to the attention supervised model, but no additional work is required to prepare training data for attention supervision.

There is still a gap between the proposed method and the attention supervised model for some prompts. Considering the pseudo attention as supervisory data is generated simply based on the edit distance between answers and key elements, a carefully constructed method to generate attention can further benefit. We will explore this problem in our future work.

References

Attali, Y. and Burstein, J. (2006). “Automated Essay Scoring with E-rater® V.2.” The Journal of Technology, Learning and Assessment, 4 (3).

Burrows, S., Gurevych, I., and Stein, B. (2015). “The Eras and Trends of Automatic Short Answer Grading.” International Journal of Artificial Intelligence in Education, 25 (1), pp. 60–117.

Cohen, J. (1968). “Weighted Kappa: Nominal Scale Agreement Provision for Scaled Disagreement or Partial Credit.” Psychological Bulletin, 70 (4), p. 213.

Dzikovska, M. O., Nielsen, R. D., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I., and Dang, H. T. (2013). “Semeval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge.” Tech. rep., University of North Texas.

Funayama, H., Sasaki, S., Matsubayashi, Y., Mizumoto, T., Suzuki, J., Mita, M., and Inui, K. (2020). “Preventing Critical Scoring Errors in Short Answer Scoring with Confidence
Wang et al. Data Augmentation by Rubrics for Short Answer Grading

Estimation.” In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pp. 237–243.

Heilman, M. and Madnani, N. (2015). “The Impact of Training Data on Automated Short Answer Scoring Performance.” In Proceedings of the 10th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 81–85.

Kingma, D. P. and Ba, J. (2014). “Adam: A Method for Stochastic Optimization.” arXiv preprint arXiv:1412.6980.

Kudo, T. (2006). “Mecab: Yet Another Part-of-speech and Morphological Analyzer.” http://mecab.sourceforge.jp.

Leacock, C. and Chodorow, M. (2003). “C-rater: Automated Scoring of Short-answer Questions.” Computers and the Humanities, 37 (4), pp. 389–405.

Levenshtein, V. I. (1966). “Binary Codes Capable of Correcting Deletions, Insertions, and Reversals.” Soviet Physics Doklady, 10 (8), pp. 707–710.

Magooda, A. E., Zahran, M. A., Rashwan, M., Raafat, H. M., and Fuyek, M. B. (2016). “Vector Based Techniques for Short Answer Grading.” In FLAIRS Conference, pp. 238–243.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). “Distributed Representations of Words and Phrases and Their Compositionality.” In Advances in Neural Information Processing Systems, pp. 3111–3119.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). “Distant Supervision for Relation Extraction without Labeled Data.” In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 1003–1011.

Mizumoto, T., Ouchi, H., Isobe, Y., Reisert, P., Nagata, R., Sekine, S., and Inui, K. (2019). “Analytic Score Prediction and Justification Identification in Automated Short Answer Scoring.” In Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 316–325.

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 ACL, pp. 752–762. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). “BLEU: A Method for Automatic Evaluation of Machine Translation.” In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics.

Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). “WordNet: Similarity: Measuring the Relatedness of Concepts.” In Demonstration Papers at HLT-NAACL 2004, pp. 38–41.
Association for Computational Linguistics.

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 10th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 97–106.

Riedel, S., Yao, L., and McCallum, A. (2010). “Modeling Relations and Their Mentions without Labeled Text.” In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 148–163. Springer.

Riordan, B., Horbach, A., Cahill, A., Zesch, T., and Lee, C. M. (2017). “Investigating Neural Architectures for Short Answer Scoring.” In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 159–168.

Sakaguchi, K., Heilman, M., and Madnani, N. (2015). “Effective Feature Integration for Automated Short Answer Scoring.” In Proceedings of NAACL, pp. 1049–1054.

Shermis, M. D., Burstein, J., Higgins, D., and Zechner, K. (2010). “Automated Essay Scoring: Writing Assessment and Instruction.” International Encyclopedia of Education, 4 (1), pp. 20–26.

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, pp. 1070–1075.

Sychev, O., Anikin, A., and Prokudin, A. (2020). “Automatic Grading and Hinting in Open-ended Text Questions.” Cognitive Systems Research, 59, pp. 264–272.

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, pp. 1882–1891.

Wei, J. and Zou, K. (2019). “Eda: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks.” arXiv preprint arXiv:1901.11196.

Ye, Z.-X. and Ling, Z.-H. (2019). “Distant Supervision Relation Extraction with Intra-bag and Inter-bag Attentions.” arXiv preprint arXiv:1904.00143.

Tianqi Wang: received a B.S degree in information security from Harbin Institute of Technology in 2012, and an M.S. degree in system informatics from Tohoku University 2016, respectively. He is currently a Ph.D. student at Graduate School of Information Science, Tohoku University. His main research interests are natural language processing and machine learning.
Hiroaki Funayama: is a first-year master’s student at the Graduate School of Information Sciences, Tohoku University. He received a B.S. degree in engineering from Tohoku University, Japan, in 2020. His research interests include natural language processing, especially in automatic scoring and other educational applications of natural language processing.

Hiroki Ouchi: received his M.S. and Ph.D. degrees from Nara Institute of Science and Technology, in 2015 and in 2018, respectively. Currently, he is a postdoctoral researcher at RIKEN Center for Advanced Intelligence Project (AIP). His research interests cover natural language processing and machine learning.

Kentaro Inui: is a Professor of the Graduate School of Information Sciences at Tohoku University, where he is head of the Natural Language Processing Lab. He also leads the Natural Language Understanding Team at the RIKEN Center for the Advanced Intelligence Project. He currently serves as Vice-chairperson of Association for Natural Language Processing, Member of Science Council of Japan, and Director of NPO FactCheck Initiative Japan.

(Received August 1, 2020)
(Revised November 8, 2020)
(Accepted December 21, 2020)