Incorporating Rubrics to Short Answer Grading

Tianqi Wang[1,2] Hiroaki Funayama[1,2] Hiroki Ouchi [2] Kentaro Inui[1,2]
[1]Tohoku University [2]RIKEN Center for Advanced Intelligence Project
{outenki,hiroaki,inui}@ecei.tohoku.ac.jp {hiroki.ouchi}@riken.jp

1 Introduction

Assignments are crucial for assessing students' knowledge in a pedagogical setting, where students are generally required to answer questions in relatively short texts. Figure 1 shows a typical example where students are asked to explain a sentence appearing in an essay to test their reading comprehension. A reference answer and rubrics are provided to show the rules for grading assignments.

Due to the diversity of open-ended questions in answer scoring, educators' workload can be excruciatingly time-consuming. 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 a response to a given prompt. The task has been studied mainly with machine learning-based approaches, focusing on exploring better representations of answers. The recently proposed neural models have been yielding strong results [1, 2].

SAG is generally a low-resource task, suffering from the scarcity of training data. Thus, it is difficult to train SAG models from answers only, leading to a poor performance. To improve the performance of SAG models especially for low-resource settings when the available training data is limited, it is easy to have the idea to incorporate rubrics to help train SAG models. However, the question is still unclear how to incorporate rubrics into neural SAG models.

In this paper, we explore two approaches to introduce rubrics to SAG models. We first augment training data with rubrics. We extract keyphrases that are highly related to the scores of answers from rubrics, and calculate the weights of words in answers based on span-wise alignments between answers and the extracted keyphrases. Highly weighed words are used as pseudo cues for scoring. Another approach to incorporate rubrics is rule-based models, scoring answers following rules defined by rubrics[3]. Thus we also explore some simple rule-based methods for comparison. The experimental results demonstrate that the performance is improved by the augmented data, especially in a low-resource setting.

2 Task and evaluation

We evaluate the SAG models on two analytical assessment tasks formalized by Mizumoto et al. [2]: 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, an SAG model is expected to output either scores [4] or labels. [5]. Given a student answer that consists of n tokens, $t^{ans}_{1:n} = (t^{ans}_1, t^{ans}_2, ..., t^{ans}_n)$, the goal is to predict the analytic score $s \in \mathbb{R}$. As shown in Figure 1, 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.). To evaluate the precision of analytic score prediction, we use quadratic weighted kappa (QWK) [6], 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 number of points of the analytic score. An example of a justification cues for item D is shown in Figure 1. As manually annotated justification cues, the phrase 西洋では in the answer refers to 西洋（では）, and 言葉を尽くして refers to 言葉を尽くして in the rubric. 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 $1 \le a \le b \le n$. In this paper we evaluate the performance on justification identification with F1 score, following the method used by Mizumoto et al [2] for evaluation.

3 Proposed method

3.1 Key idea

Question: 傍線部(1)「こうした緊張したスタンスこそが饒舌な西洋文化を導いてきた」とあるが、それはどういうことですか。句読点とも七〇字以内で説明せよ。

Reference answer: (B)他人を自分とは異質な考え方を持つ人間と見なす(A)西洋では、(C)自分の意見に同意を得るために、(D)言葉を尽くして他人を説得する技術が培われたということ。

Rubric:

(A)「西洋（では）」（＝話題の中心）…… 2点
- 「ヨーロッパ」も可。「外国」は不可。
- 「西洋人は～」「ヨーロッパの人は～」なども可。

((B) と (C) 略)

(D)「言葉を尽くして他人を説得する」…… 6点
- D①: 「言葉を尽くして」……3点
  - 「手を替え足を替え」「あれこれ工夫して」（＝「言葉」抜け）はD①ポイント2点。
  - 「饒舌」そのままは、D①ポイント1点。
- D②: 「（他人を）説得する」……3点
  - 「（自分の考えを）説明（する）」「（自分の考えを）伝えようとする」は、D②ポイント1点。

Figure 1: A typical example of a prompt to a provided essay to test the reading comprehension of students. A reference answer and rubrics for scoring are also shown. Texts in green are manually annotated justification cues.

To clarify our key idea, we first introduce attention unsupervised baseline model (ATT-UNSUP) [1] and an attention supervised model (ATT-SUP) [2] for comparison. As illustrated in Figure 2(a), the ATT-UNSUP model encodes student answers with an embedding layer and a Bi-LSTM layer. Then the weighted sum of hidden states is input to a regression layer for prediction of score. The weights are learned by a attention layer on top of the Bi-LSTM layer. We take the ATT-UNSUP model as the baseline in this paper. With additional annotation on justification cues, the ATT-SUP model is trained with gold attention supervisory signals, as is shown in Figure 2(b). Attention to tokens of justification cues is set to 1, and attention to other tokens is set to 0. It is reported that the performance is improved by attention supervision [2]. In this paper we list the performance of ATT-SUP model for reference.

Rubrics define key elements with keywords or keyphrases as examples. **Key elements** are concepts or information defined by the rubrics, that an answer needs to contain in order to receive points. As is shown in Figure 1, the phrase 西洋では is a keyphrases to define key element for item A, so answers will receive 2 points if they contain a similar phrase. Our key idea is illustrated in Figure 2(c). Instead of manually supervising justification cue signals, we consider using the rubrics to automatically identify which span of a given answer should be attended.

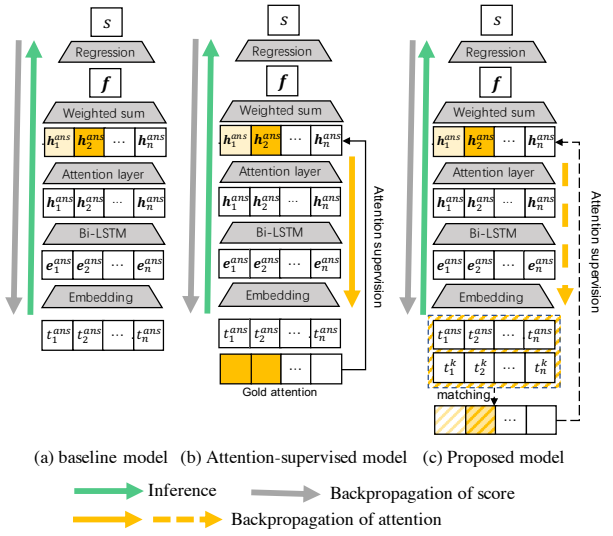

Figure 2: The architecture of baseline model, attention supervised model and proposed method.

If a span that is likely to be a justification cue is identified in a training instance, we then use it as a pseudo supervisory signal to train the attention prediction component. Otherwise we do not use that instance for supervising attention prediction, but use only its gold score to train the model. The output of regression layer is the predicted score, and the output of attention layer is considered as prediction of justification cues.

3.2 Data augmentation

We generate pseudo attention supervisory signals based on span-wise matching to rubrics. A concrete example is given in Figure 3. Answers and rubrics are tokenized by MeCab [7], and keyphrases such as（他人を）説得する are extracted from the given rubrics. We then identify justification cues in a given answer by matching spans to the keyphrases.

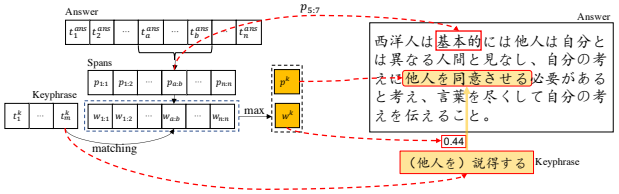

Figure 3: Match justification cues.

Given an answer consisting of n tokens $t^{ans}_{1:n} = (t^{ans}_1, \ldots, t^{ans}_n)$, we first enumerate all possible spans: $\mathcal{P} = \{a{:}b | 1 \le a \le b \le n\}$, where each span $p_{a:b} = (t^{ans}_a, \ldots, t^{ans}_b)$ is a subsequence of the tokens in the answer. Each keyphrase is a sequence of m tokens:

$t^k_{1:m} = (t^k_1, \ldots, t^k_m)$. The similarity score $w_{a:b}$ between a keyphrase $t^k_{1:m}$ and each span $p_{a:b}$ is calculated $w_{a:b} = \mathrm{sim}(t^k_{1:m}, p_{a:b})$. Based on the similarities, we define the matching score w^k, how well the keyphrase $t^k_{1:m}$ matches (or is entailed in) the answer, and extract the span with the highest similarity as the pseudo justification cue: $p^k = \mathrm{argmax}_{p_{a:b} \in \mathcal{P}} w_{a:b}$.

Based on the pseudo justification cues (spans) p^k, we define a pseudo supervisory signal $\alpha^{\mathrm{pseudo}}_i$ of each token t^{ans}_i in an answer $t^{ans}_{1:n}$ to learn the attention mechanism, instead of the ground-truth one. Specifically, if a token t^{ans}_i is contained in a pseudo justification cue $p^k = (t^k_a, \ldots, t^k_b)$, we use the similarity score w^k as the pseudo supervisory signal. We pick up the highest one of the similarities if a token is contained in multiple extracted spans.

3.3 Attention semi-supervised learning

We propose an attention semi-supervised method to train the SAG model. Because not all of the pseudo justification cues are reliable, only high enough attention values to justification cues will be used as pseudo signals for attention supervising. To filter out the unreliable pseudo justification cues, a binary label η_j is attached to each answer a_j. The value of η_j is set to 1 if $max(\alpha^{pseudo}_j) \geq T$, otherwise the value is set to 0. The threshold T is a hyperparameter. The answer a_j is considered as matched to justification cues only when $\eta_j = 1$. Besides, in order to enhance the effectiveness of attention supervision, we binarize the labeled attention $\boldsymbol{\alpha}^{\mathrm{pseudo}}$ to $\boldsymbol{\beta}^{\mathrm{pseudo}}$. For the i^{th} token of answer a_j, the attention value $\beta^{pseudo}_{j,i}$ is set to 1 if $\alpha^{pseudo}_{j,i} > T$, and is set to 0 otherwise.

The loss function is designed as follows:

$$l = \frac{1}{N}\sum_{j=1}^{N}(s^{\mathrm{gold}}_j - s_j)^2 + \frac{1}{Nn}\sum_{j=1}^{N}\sum_{i=1}^{n}(\eta_j\beta^{\mathrm{pseudo}}_{j,i} - \eta_j\alpha_{j,i})^2$$

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.

3.4 Rule-based models

Another approach to incorporate rubrics to SAG is rule-based methods. As described in Figure 1, 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 any keywords are contained by the answer, and the sum of scores assigned in each step is output as the final score for the answer. We proposed two simple rule-based models that score answers the rules.

Rule-based model on top of regexp matching (**RGEXP-RULE**): We create regular expressions for keyphrases for each step based on the rubrics. Then we identify keyphrases in a given answer by regexp matching for each step, and assign points to the answer for each matched keyphrases. The sum of corresponding scores of matched keyphrases in each step is output as the prediction of answer score. Only the keyphrases with the highest score will be selected if multiple keyphrases are matched in one step.

Rule-based model on top of span-wise matching (**SPAN-RULE**): This method works similarly to **RGEXP-RULE** introduced above, but the span-wise matching introduced in Section 3.2 is used instead of regexp matching. We apply different values of T from 0.1 to 1.0, and take the value leading to the best performance over all prompts.

4 Experiments

We apply our proposed method on the dataset provided by Mizumoto et al. [2] [1)]. The dataset includes 6 prompts (*Q1*~*Q6*), with 1600 answers as training data, 250 answers as development data, and 250 answers as test data for each prompt. To focus on the performance in low-settings, we train the models on various sizes of training data, ranging from 50 to 200, and randomly select 30 develop data.

The embedding layer is initialized wiyh a 100-dimensional Word2Vec word embeddings [8] pre-trained on Japanese Wikipedia data. We freeze the embedding layer during the training phase to reduce the number of parameters to learn. The dimension of the Bi-LSTM layer is set to $d^{lstm} = 250$, and the dropout probability is set to 0.5. The parameter d for $\boldsymbol{W}$ and $\boldsymbol{m}$ of the attention layer is set to 100. $\boldsymbol{W}$ and $\boldsymbol{m}$ are initialized randomly from the normal distribution, with the standard deviation set to 0.01.

To generate pseudo attention supervisory signals, Levenshtein edit distance [9] is used to match answer spans to keyphrases. The hyperparameter T is selected based on develope dataset. In this paper, we train our model on each item (21 items in total). The parameters are optimized with the Adam [10] optimizer, with learning rate of 0.001.

1) https://aip-nlu.gitlab.io/projects/sas-j

Experiments are repeated for 5 times with different random seed from 0 to 4 for initialization, and the average results over all random seeds are used as the final results.

5 Results and discussion

Table 1: QWK on analytica score prediction.

Training size	50	100	200
ATT-UNSUP (base)	0.644	0.703	0.78
REGEXP-RULE	0.353	0.353	0.353
SPAN-RULE	0.510	0.510	0.510
OUR	**0.678**	**0.740**	**0.783**
ATT-SUP (ref)	0.681	0.743	0.794

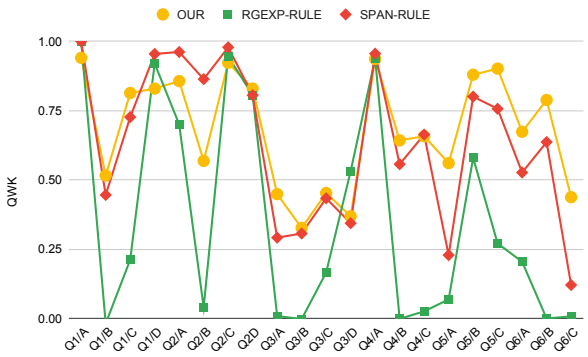

Figure 4: QWK analytic score prediction on each prompt. The size of training data for OUR model is 50.

The experimental results are listed in Table 1 and Table 2, corresponding to QWK for analytic score prediction and F1 score for justification identification. Again, the ATT-UNSUP model is the baseline model, and ATT-SUP with manually annotation on justification cues as attention supervisory signals is for reference.

The performance of rule-based models is very limited, as is shown in Table 1. To figure the reason, we show the performance on each prompt in Figure 4. Note that the rule-based models performance well for some specific prompts, where the justification cues are easy to match, and the rules for scoring is simple. However, for prompts where the rubrics are complicated, it is more challenging to identify justification cues, leading to a extremely low performance. The span-wise matching method performs better than regexp matching, and OUR neural model outperforms the rule-based models overall. This demonstrates that a well designed matching method is necessary for rule-based method to incorporate rubrics to SAG models, and the neural network helps make a stable model, especially on prompts with complicated rubrics. We will explore this problem further in our future work.

Table 1 also indicates that compared to the ATT-UNSUP model, the performance on analytic score prediction is improved by data augmentation. The performance of OUR model is comparable to the ATT-SUP model trained with manually annotated justification cues, but no additional annotation data is required by our model. 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. The F1 score for justification identification is also improved through various training sizes, as shown in Table 2, indicating better interpretability on the scoring results. However there is still a gap between OUR model and ATT-SUP model. That indicates that there is still much room for improvement with a more carefully designed method to generate pseudo justification cues.

Table 2: F1 score on justification identification.

Training size	50	100	200
ATT-UNSUP	0.320	0.331	0.307
OUR	**0.546**	**0.612**	**0.611**
ATT-SUP (ref)	0.625	0.677	0.700

6 Conclusion

Short Answer Grading task plays an important role in education context. However it suffers from the scarcity of training data, making its application limited. To improve the performance of SAG models, we explored approaches to incorporate rubrics to the SAG task. The rule-based models work well on simple prompts, but the performance is limited for prompts with complicated rubrics. The data augmentation with rubrics improves the performance on both analytic score prediction and justification identification compared to the attention unsupervised baseline model, especially in low-resource settings. Considering the pseudo justification cues are generated by a simple span-wise matching method, a more carefully constructed method can lead to further benefits. We will explore this issue in our future work.

References

[1] Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chong Min Lee. 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, 2017.

[2] Tomoya Mizumoto, Hiroki Ouchi, Yoriko Isobe, Paul Reisert, Ryo Nagata, Satoshi Sekine, and Kentaro Inui. Analytic score prediction and justification identification in automated short answer scoring. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 316–325, 2019.

[3] Lakshmi Ramachandran, Jian Cheng, and Peter Foltz. 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*, pp. 97–106, 2015.

[4] Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 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, 2011.

[5] Myroslava O Dzikovska, Rodney D Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa T Dang. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. Technical report, NORTH TEXAS STATE UNIV DENTON, 2013.

[6] Jacob Cohen. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, Vol. 70, No. 4, p. 213, 1968.

[7] Taku Kudo. Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab. sourceforge. jp*, 2006.

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[9] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10, pp. 707–710, 1966.

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