Automatic Short Answer Grading with Rubric-based Semantic Embedding Optimization

Bo Wang[1,a)] Tsunenori Ishioka[2,b)] Tsunenori Mine[1,c)]

Abstract: Large-scaled encoders such as BERT have been actively used for sentence embedding in automatic scoring. However, the embedding may not be optimal due to non-uniform vector distribution. By conducting fast contrastive learning, methods like SBERT got better semantic embeddings and were actively used in textual similarity datasets. However, the cost to obtain the similarities limits its application to automatic grading. In this paper, we propose a method of calculating similarity from the rubric to perform contrastive learning for a better semantic embedding. We conducted extensive experiments on 60,000 answer/question data for three independent questions. The experimental results show that the proposed method outperforms all baselines in terms of accuracy and computation time.

Keywords: Automatic grading, Rubric information, Semantic embedding, Contrastive learning.

1. Introduction

General neural network-based automatic grading methods usually follow a sentence embedding - regression/classification workflow. Recently, thanks to the great encoding ability of large-scale pre-trained encoders such as BERT (Bidirectional Encoder Representations from Transformers [4]), their applications in automatic grading are also growing rapidly. To get better performance, many BERT-based automatic grading methods would 'pre-train' the BERT again with texts in the related fields, then finetune or connect BERT to downstream network for grading with the specific task dataset.

However, [5], [8], [17] pointed out that the word vectors presented by BERT are not evenly distributed, which also hugely influences the quality of the sentence vectors. The training procedure also takes a long time. More importantly, as a unique feature of the grading dataset, all the answers are to the same question so there exists some similarity relationships between them, even the scores are different. However, neither of the pre-train and finetune method can capture the similarity information well.

As a solution to BERT embedding problem, [13] proposed to finetune BERT with a fast contrastive learning procedure (known as Sentence-BERT, SBERT). In short, it used the similarity regression or triplet objective function to make the sentence embeddings of 'similar sentences more closer and dissimilar ones further apart', thus making the embedding distribution more uniformed with a shorter training time, and got better results than normal BERT training methods in textual similarity datasets.

Because of the ability to make the semantic embedding better, an increase in the usage of SBERT as an encoder has been observed, where [3], [11], [12] used pre-trained SBERT directly and got good results. It can be expected that better results should be achieved if further contrastive learning procedure on the grading dataset can be performed. However, to the best of our knowledge, no such work or method has been proposed. This is partly because (i) the previous similarity definitions between answers did not take into account the various criteria for score calculation, and were often used for final scoring only, (ii) the method of how to define the similarity reasonably is not clear, either.

To make the similarity relationship between answers fully exploited with contrastive learning, so that better embeddings of answers can be obtained to perform better grading, we propose a simple but efficient method to perform semantic embedding optimization on BERT for automatic grading:

(1) As a simplest thought, we define the similarity of two answers as the ratio of their scores and construct the answer-pairs, which makes the similarity regression objective function become available for grading dataset.

(2) Many questions are scored with several correct conditions, but the score assignment is not usually linear[*1], this makes the score-based similarity may deviate from the ideal, content-related similarity. To solve this problem, we propose to use the information from the rubric, to correct the similarity as the ratio of the number of conditions satisfied by two answers respectively. We also add some tolerances to make the similarity calculation more robust.

The overall procedure of our method is shown in Fig.1. To

[1] Kyushu Universtiy, Fukuoka City, Fukuoka 819-0395, Japan
[2] The National Center for University Entrance Examinations, Meguro-ku, Tokyo 153-8501, Japan
[a)] wangbo.rw@gmail.com
[b)] tunenori@rd.dnc.ac.jp
[c)] mine@ait.kyushu-u.ac.jp

[*1] that is, points are usually not assigned as the 'one point for each correct condition' rule, we will give specific examples on Sec.3.3.

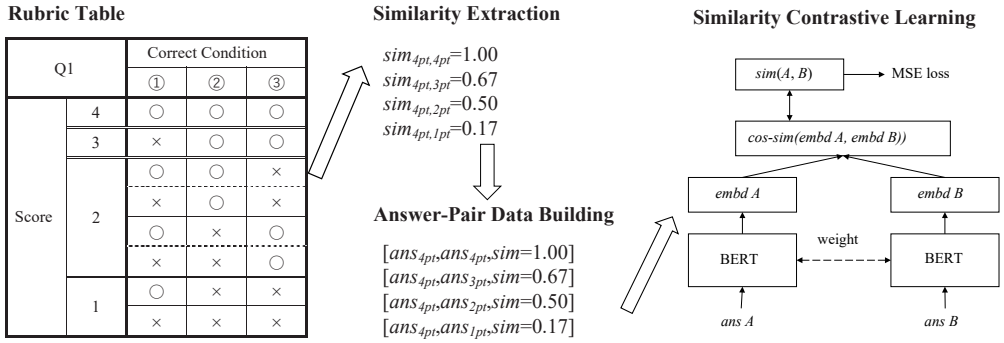


Fig. 1 Full procedure of rubric-based semantic embedding optimization

evaluate the methods, we used three independent question grading datasets with sentence vector analysis and grading errors. The results demonstrate a better embedding distribution and efficiency of our method over competitive baselines. The main contributions of this paper are:

1. We observed that different answers to the same question often have similar relationships and proposed a simple score-based similarity definition and sentence-pair construction method, which makes the contrastive learning on grading dataset available, so that better semantic embeddings can be obtained.
2. We further proposed to extract similarity information from rubric and added slight tolerances to make further corrections to the similarity.
3. The proposed method is simple and easy to use, which overperformed the traditional BERT pre-train methods in terms of both efficiency and results.

2. Related Works

Since the release of BERT, there have been many studies using BERT for automatic grading. [7] used hypernym and synonym to replace words in the answers to expand data, then fed the sentences to a pre-trained BERT to obtain word embeddings and used a BiLSTM for grading. [9] used several oversampling technologies such as back-translation, swap contents and finetuned BERT with oversampled answers. [16], [18] used unlabeled test-related corpus to pre-train BERT again, then finetuned BERT with the training data.

However, the vector distribution of BERT is not uniformly distributed, where [8], [13], [15] proposed their corresponding methods (known as SBERT, BERT-flow and BERT-whitening) to optimize BERT for a better semantic embedding. As a simpler finetuning method with contrasitve learning which can be used immediately without additional matrix transformations while significantly speed-up the training, there is a trend to use the SBERT for sentence embedding on automatic grading.

For example, [2], [11], [14] used standard or randomly selected answers of each score as the reference, then fed them into the pre-trained SBERT along with the test set to calculate the vector similarity and determined the score. [12] used a multilingual pre-trained SBERT to obtain the sentence embeddings and used hyperparameter searching for score prediction. [3], [6] tried different combinations of inputs (e.g., answer, answer plus question, etc.) to obtain embeddings from SBERT, followed by a multinomial regression method or additional encoder blocks for grading, respectively.

However, the above methods mostly used the pre-trained SBERT directly or trained SBERT back as the old ways to train BERT. When it comes to the other language like Japanese, as there are only two Japanese SBERTs pre-trained with limited datasets, which are not comparable to the huge pre-train corpus of English SBERTs, the grading effect may become much worse when using the model directly. So it is meaningful and necessary to explore how to conduct contrastive learning on BERT to optimize the semantic embeddings for a better automatic grading.

3. Rubric-Based Similarity Semantic Embedding Optimization

3.1 Preliminary: Contrastive Learning and Sentence-BERT

The contrastive learning procedure means bringing sentences with similar semantic meanings closer together and pulling those with more different meanings further apart. As a simple implementation, SBERT [13] used siamese/triplet networks to finetune BERT with contrastive learning using similarity regression/triplet objective function. As a result, SBERT obtained more aligned and uniformed sentence embeddings in the vector space in a much shorter training time and achieved good results on textual similarity datasets.

Here, we introduce the siamese networks and corresponding similarity regression objective function as the preliminary, which is then used in our method[*2].

Similarity Regression Objective Function

This objective function is originally used for the text similarity datasets with sentence pairs and the similarity information provided (e.g., [*sent 1, sent 2, similarity*], such as STS[1], SICK [10] dataset[*3]). Specifically, two sentences in the pair are fed into BERT separately to get the sentence embeddings, then cosine vector similarity between them is calculated. The BERT network is optimized with a mean squared-error (MSE) loss function between the calculated similarity and real similarity, as shown in Fig.1 right part.

[*2] The triplet objective function will also be introduced as a baseline in Sec.4.2.

[*3] STS: Semantic Textual Similarity Dataset, SICK: Sentences Involving Compositional Knowledge Dataset

$$\mathcal{L}_{similarity} = \frac{1}{n}\sum^{n}(sim_{12} - sim(embd(sent_1), embd(sent_2)))^2 \quad (1)$$

The similarity regression objective function is clearly better than the triplet if can be used to the grading dataset, for it takes the similarity relationship between answers into consideration. However, till now no works are proposed to use this objective function to finetune BERT further for automatic grading, whose biggest difficulty lies on the lack of similarity information between two answers in grading datasets, and also the construction method of sentence-pairs.

3.2 Score-Based Similarity Definition and Answer-Pair Building

To make contrastive learning available, the similarity of the two answers needs to be determined. Here, we first propose a simple idea which uses the ratio of the scores of two answers as the similarity.

Definition 1 (score-based similarity)**.** *Suppose there are two answers $\boldsymbol{x}$ and $\boldsymbol{y}$ to the same question $\boldsymbol{Q}$, their scores are $sco_{\boldsymbol{x}}$ and $sco_{\boldsymbol{y}}$, respectively, then the similarity is defined as:*

$$sim^{sco}_{\boldsymbol{x},\boldsymbol{y}} = \frac{sco_{\boldsymbol{x}}}{sco_{\boldsymbol{y}}} \quad (2)$$

As for the answer pairs building, to let the network learn all the pairwise similarity information without redundancy, we constructed answer pairs with the highest scored answer as the reference to others (including itself). For example, for a grading dataset with scores ranging from 1 to 4, we build the answer pairs as $[ans_{4pt}, ans_{4pt}, sim = 4/4], ..., [ans_{4pt}, ans_{1pt}, sim = 1/4]$.

3.3 Extract Similarity Information from Rubric

If possible, an ideal and reasonable textual similarity of two texts should be defined and determined as the ratio of 'the number of contents (features) that overlaps'. Although the semantic embedding optimization can be conducted after introducing the score-based similarity, this similarity, however, cannot always meet the ideal definition.

This is because in real question-answer grading, there is often more than one correct condition to consider to determine the score, while the score-assignment standard is usually not linear (i.e., points are not assigned as the 'one point for each correct condition' rule). Therefore, the ratio of scores between two answers is not always proportional to the ratio of 'the number of contents that overlaps', thus make the score-based similarity not very accurate to the real similarity.

As it is also impractical to separate the contents of answers to further calculate similarity, here we propose to use the grading table, or known as 'rubric' to extract information and define the similarity. We choose the rubric table because it is easy-to-get when grading and contains information about the conditions that each level's answers must meet, which typically fits the concept of 'the number of contents that overlap' of an ideal similarity definition.

Formally, suppose for question Q, n correct conditions are considered to determine the final score, i.e. $Con = \{c_1, c_2, ..., c_n\}$. Then the condition satisfaction situation of an answer $\boldsymbol{a}$ is an arrangement of 0, 1 of Con, e.g. $Con_{\boldsymbol{a}} = \{c_1 = 1, c_2 = 0, ..., c_n = 1\}$, where $c_i = 1$ means condition i is satisfied and vice versa.

Generally, rubrics are used to assign scores in accordance with the condition-score rules listed, that is, there is a function $f : Con \rightarrow sco$ to determine the score. But the inverse procedure is also possible, which is $g : sco \rightarrow \boldsymbol{Con}, Con \in \boldsymbol{Con}$, where our rubric-based similarity definition also starts from here[*4].

Definition 2 (rubric-based similarity)**.** *The similarity of answer $\boldsymbol{x}$ and $\boldsymbol{y}$ with score $sco_{\boldsymbol{x}}$ and $sco_{\boldsymbol{y}}$ is defined as:*

$$sim^{rub}_{\boldsymbol{x},\boldsymbol{y}} = \frac{||g(sco_{\boldsymbol{x}})||}{||g(sco_{\boldsymbol{y}})||} = \frac{||\boldsymbol{Con_x}||}{||\boldsymbol{Con_y}||} = \frac{avg(||Con_{\boldsymbol{x}^1}|| + ||Con_{\boldsymbol{x}^2}|| + ...)}{avg(||Con_{\boldsymbol{y}^1}|| + ||Con_{\boldsymbol{y}^2}|| + ...)} = \frac{avg(\sum_{i=1}^{n} c_i^1 + \sum_{i=1}^{n} c_i^2 + ...), c_i^p \in Con_{\boldsymbol{x}^p}}{avg(\sum_{j=1}^{n} c_j^1 + \sum_{j=1}^{n} c_j^2 + ...), c_j^q \in Con_{\boldsymbol{y}^q}} \quad (3)$$

The effect of using this definition to obtain a more reasonable similarity is obvious. As an example, for the grading question $Q1$, whose rubric table is shown in Fig.1 left part (we will further introduce it in Sec.4.1), if we use the score-based similarity, the similarities of 4-point answer to 3-point answer, and to 1-point answer are $3/4 = 0.75$ and $1/4 = 0.25$ respectively. But if we used the rubric-based definition, they are $2/3 = 0.67$ and $(avg(1 + 0))/3 = 0.17$, which clearly show a more realistic situation.

In addition, consider the fact that even answers with the same score may be slightly different in semantic expression, we propose to add some tolerances after obtaining the similarity in Def.2 as a correction term. In the experiments, we tested two tolerances *toler.*: 5% and 10% this time.

Definition 3 (corrected rubric-based similarity)**.** *The similarity of answer $\boldsymbol{x}$ and $\boldsymbol{y}$ with score $sco_{\boldsymbol{x}}$ and $sco_{\boldsymbol{y}}$ is defined as:*

$$sim^{tol}_{\boldsymbol{x},\boldsymbol{y}} = sim^{rub}_{\boldsymbol{x},\boldsymbol{y}} - toler. \quad (4)$$

3.4 Automatic Grading with Refined Semantic Embedding

After the embedding optimization finished with our built sentence-pairs, the sentence embeddings were obtained from the trained SBERT. Here a simple 5-MLP was used to perform automatic grading with the sentence embeddings.

4. Experiments

We conducted experiments on three independent questions of short-text grading and compared the grading error and training speed of each method. We also made a vector analysis for the obtained sentence embeddings.

4.1 Datasets

We used the dataset of short essay questions from Japanese subject, the trial test for Japanese common university entrance examination in 2018. It has three questions corresponding to three articles. Students were asked to answer the meaning of the underlined sentences after reading each article. The three questions

[*4] The reason why we did not use the inverse function of f directly is because when more than one situation can be evaluated to the same score, f has no inverse function.

Table 1 Basic information for the datasets

	Num. of total data	Word Limit	Data id for pre-train/ exp.	Data dist. of 1–10k (1,2,3,4pt)	Num. of built sent-pair	Actual num. of data used
Q1	67,745	<30	ids 1–10k/ ids 10k–20k	73 2,354 3 7,570	5,300	7,138
Q2		<40		987 4,163 1 4,849	6,135	8,260
Q3		80–120		5,604 2,516 245 1,635	6,212	6,784

Q1 question	「指差しが魔法のような力を発揮する」とは、どういうことか。三十字以内で書け。
	(What does "Pointing the finger has a magical power" means? Answer within thirty words.)
Std ans 1	ことばを用いなくても意思が伝達できること。
	(The ability to communicate without using words.)
Std ans 2	指さしによって相手に頼んだり尋ねたりできること。
	(The ability to ask someone for help by pointing fingers.)
Correct con. ①	Written in 30 words.
Correct con. ②	Had mentioned 'without using words' or 'by using pointing fingers'.
Correct con. ③	Had mentioned 'to make communication' or 'getting attention of other people'.

Q1 Rubric		Correct Condition ①	②	③
Score	4	○	○	○
	3	×	○	○
	2	○	○	×
		×	○	×
		○	×	○
		×	×	○
	1	○	×	×
		×	×	×

Fig. 2 The question, reference answers, correct conditions and rubric table for $Q1$

are all originally scored with *a,b,c,d* 4 ranks, which we simply referred as *4,3,2,1* point(s) respectively. As an example, the question text, reference answers, correct conditions and rubric of $Q1$ is shown in Fig.2, the general information of datasets and built sentence-pairs is shown in Table.1.

Although each grading dataset has more than 60k pieces of answers, as in our pre-experiment, to get the best result, traditional BERT-based method needed more than 5 hours for 10k data/question. Considering the time cost, we used the answers with ids 1–10k to pre-train the BERT for traditional methods as baselines, and also used them as the source to build the sentence-pair needed for our embedding optimization method. The answers with ids of 10k–20k were used as the experimental data, where 5-fold cross-validation experiments were conducted to verify the effect. An additional experiment with an amount of 5-times data (the full dataset) was also performed only with the SBERT-based and our proposed methods.

4.2 Baselines

We used the following competitive methods as our baselines.

- ***Traditional BERT pre-train/finetune-based methods***:
  They represents the traditional usage of BERT automatic grading. ***BERT-lstm*** [7] used the pre-trained BERT directly to obtain the word vector and connected it to a BiSLTM for the score prediction, while ***BERT-pre-lstm*** pre-trained BERT with answers of ids 1–10k and connected it to the BiLSTM network. ***BERT-pre-fine*** pre-trained BERT, then finetuned it on the experimental dataset directly with answer ids 10k–20k for grading.
- ***Pre-trained SBERT-based methods***:
  Here, the only two Japanese pre-trained-SBERTs, *'jp-sbert-colorfulscoop'*[*5] and *'jp-sbert-sonosia'* [*6] (as ***SBERT scoop, SBERT sono***), can be found and were used for obtaining the sentence embeddings for score prediction.
- ***Triplet Objective-based method***:
  As introduced in Sec.3.1, SBERT has a triplet contrastive finetuning method that does not require similarity information but only label, although it treats answers with different scores as independent categories. We also added it as a baseline ***Tri learn***. Specifically, given an anchor sentence ***anc***, a positive sentence ***pos***, and a negative sentence ***neg*** as the triplet [***anc***, ***pos***, ***neg***], the BERT network is finetuned with the triplet loss: [*7].

$$\mathcal{L}_{tri} = max(\|embd_{\boldsymbol{anc}} - embd_{\boldsymbol{pos}}\| - \|embd_{\boldsymbol{anc}} - embd_{\boldsymbol{neg}}\| + \varepsilon,\ 0) \tag{5}$$

- ***Ablation and Effect Study***:
  For ablation study, we directly let *'tohoku-jp-bert-wwm'*, the base network of our embedding optimization method, to produce sentence vectors in a SBERT way and used them to make score predictions, as ***Abl wwm***. In addition, to demonstrate our optimization effect even with fewer data, we use half or fourth of the sentence pairs for our contrastive training as an effect study, as ***Eff half*** and ***Eff fourth***.

4.3 Metrics

In this experiment, the root-mean-squared error (RMSE) between predicted scores and true scores is used as the indicator for model performance judgment.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\widehat{sco}_i - sco_i)^2}{n}} \tag{6}$$

The reason why we did not use the common metrics such as

[*5] https://huggingface.co/sonoisa/sentence-bert-base-ja-mean-tokens
[*6] https://huggingface.co/colorfulscoop/sbert-base-ja

[*7] To release the full strength of triplet learning, we built the triplet dataset using a 'handshake' approach. e.g., for $Q1$, we constructed [4(pt)-4(pt)-3(pt)], [4-4-2], [4-4-1], [3-3-2], [3-3-1], [2-2-1] six modes of triplets to let the BERT gain enough information.

Table 2 Results on $Q1, Q2, Q3$ in RMSE and the efficiency comparison. The best baseline results are underlined and the best results are shown in **bold**.

Method/RMSE	*Q1*	*Q2*	*Q3*	Time for pre-train	Time for train+pred
BERT-lstm	0.0665	0.1058	0.1065	/	59min
BERT-pre-lstm	0.0624	0.1067	0.1082	39min	
BERT-pre-fine	0.4431	0.6727	0.5578		6min5s
BERT-lstm (wwm)	0.0665	0.1058	0.1065	/	5h22min
BERT-pre-lstm(wwm)	0.0539	0.0944	0.1038	37min	
BERT-pre-fine(wwm)	0.5165	0.7304	0.5865		9min12s
SBERT-color	0.0827	0.1565	0.1476	/	28s
SBERT-sono	0.0836	0.1628	0.1469		
Tri learn	0.0576	0.0934	0.1063	4min48s	
Score-based sim	0.0514	0.0834	0.0937	3min42s	
Rubric-based sim	0.0510	0.0839	**0.0927**		
Rubric-5%toler	**0.0498**	**0.0830**	0.0959		
Rubric-10%toler	0.0501	0.0844	0.0956		
Abl wwm	0.0728	0.1400	0.1361	/	
Eff half	0.0550	0.0963	0.1057	2min12s	
Eff fourth	0.0572	0.1047	0.1156	57s	

accuracy, F1 value is because when dealing with automatic grading as classification, most methods follow a maximum likelihood principle. However, for example, if the true score is 4, while the two methods give 70% and 85% confidence for 4-point respectively, then their accuracy are the same, which cannot fully show the difference of real performance. Compared to them, as a normalized absolute error, RMSE is more suitable for performance comparison.

4.4 Implementation

For the traditional BERT-based methods, we used *'kyoto-jp-bert-base'*[*8] and *'tohoku-jp-bert-base-wwm'*[*9] (whole-word-masking) as the base encoder. The BiLSTM we used has 3 layers with 300 neurons per layer with attention enabled. For pre-trained SBERT-based methods, we extracted sentence vectors from two SBERTs as described in baseline. For our method, also the triplet learning, ablation and effect study, we all use *'tohoku-jp-bert-base-wwm'* as the base BERT network. All the BERTs were pre-trained 6 times. After the learning, we extracted the sentence embeddings of answers with ids 10k–20k from the optimized BERTs and used a 5-MLP network to perform score prediction. The the numbers of neurons each layer are (512,256,128,32,1), while ReLu was used as the activation function between layers. Finally, the BERT-finetune, BiLSTM and MLP were trained using RMSE loss. All experiments were conducted in a 5-fold cross-validation with 64% train/ 16% validation/ 20% test settings.

5. Results and analysis

5.1 RMSE Results and Efficiency Comparison

The results of our method and the baseline methods on the three questions with answers id 10k–20k are shown in Table.2 and Fig.3. The average time per question for pre-training/our embedding optimization and for the whole training are also shown for the comparison of the efficiency.

From the table, we can find that our rubric-based semantic embedding optimization model achieved the best results. After

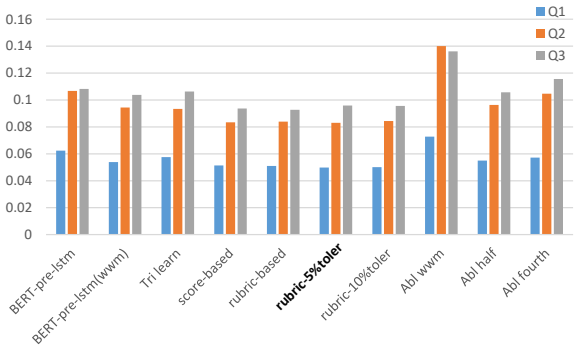


Fig. 3 Results with 10k data on $Q1, Q2, Q3$ in RMSE

the second pre-training, the traditional methods showed some improvements in results especially for ***BERT-pre-lstm(wwm)***, whose average performance had improved 8.3%. All the fine-tuned methods performed badly, showing an external network may be needed when making score predictions. The only two Japanese pre-trained SBERT models were not very effective to generate good sentence embeddings, which even obtained worse results than using untrained BERT ***Abl wwm*** in a SBERT way. In addition, although ***Tri learn*** showed good results, which is comparable to traditional BERT-based methods, it is still weaker to the simplest definition - score-based similarity of our embedding optimization method.

As for our proposed methods, we can see that it made sense to add some tolerances to the rubric-based similarity, which obtained better results in $Q1, Q2$. Meanwhile, even using half of the training data, ***Eff half*** still obtained results comparable to the best traditional ***BERT-pre-fine(wwm)*** method. The ***Eff fourth*** lost some advantages on $Q3$, which we consider is due to a longer content and the few training data (only used 2,165 answers of 10k data). However, we were much faster than the traditional methods in both pre-training and training and prediction time.

5.2 Full Dataset RMSE Results

Table.3 showed the RMSE results of the whole data experiment with SBERT-based and our models, where we observe a similar trend comparing with the results in Table.2. Although the proposed methods performed slightly worse than the past results, we

[*8] https://nlp.ist.i.kyoto-u.ac.jp/?ku_bert_japanese
[*9] https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

found no significant difference in a $p \leq 0.05$ level. Meanwhile, they are still better than traditional BERT-based results even with only 10k (in fact less than 10k) pre-train data to predict 57k answers' scores.

Table 3 Results with full 57k data on $Q1$, $Q2$, $Q3$ in RMSE

Method/RMSE	*Q1*	*Q2*	*Q3*
SBERT-color	0.0846	0.1555	0.1480
SBERT-sono	0.0825	0.1603	0.1507
Tri learn	0.0595	0.0942	0.1104
score-based	0.0518	0.0868	0.0959
rubric-based	0.0514	0.0862	**0.0950**
rubric-5%toler	**0.0507**	**0.0856**	0.0964
rubric-10%toler	0.0515	0.0863	0.0973
Abl wwm	0.0732	0.1332	0.1354
Eff half	0.0556	0.0973	0.1071
Eff fourth	0.0604	0.1051	0.1161

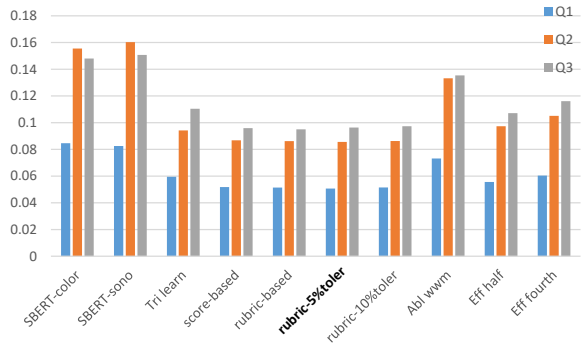
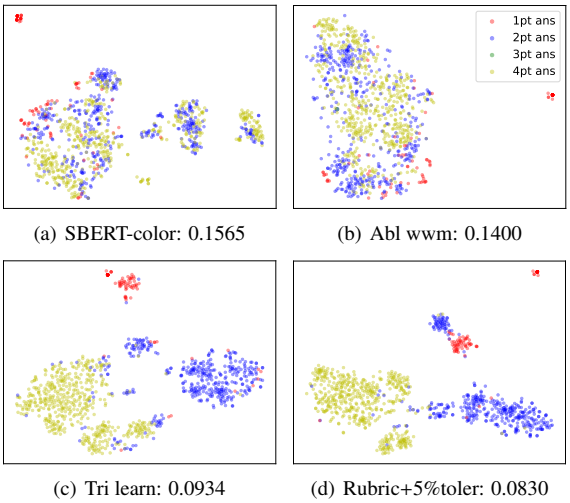


Fig. 4 Results with full 57k data on $Q1$, $Q2$, $Q3$ in RMSE

5.3 Vector Space Analysis

Finally, by using the t-SNE, the visualized distribution of the sentence vectors with ids 10,001–11,000 of $Q2$, which were obtained from our trained BERT and three baselines, are shown in Fig.5. We can find that the sentence embeddings of different



(a) SBERT-color: 0.1565 (b) Abl wwm: 0.1400

(c) Tri learn: 0.0934 (d) Rubric+5%toler: 0.0830

Fig. 5 Distribution of sentence embeddings and RMSE result on $Q2$

scores were often mixed together when using BERT or SBERT models without further embedding optimization, which burdened the later grading process. Although triplet learning well separated the sentence vectors with various scores, compared with our method, our embedded sentence vectors had clearer boundaries and more accurate divisions, which contributed to the better results.

6. Conclusion and Future Work

In this paper, we proposed a novel and efficient rubric-based semantic embedding optimization method for automatic grading. As conclusion, we found that our idea of defining the similarity as 'the ratio of the number of satisfied conditions of two answers, plus a slight semantic tolerance of 5%' achieved the best results. The superiority of our method can be found through sentence vector space analysis, where the separations and boundaries of differently scored answer vectors were very clear. Meanwhile, with a simple MLP network to make grading from obtained sentence embeddings, it achieved the best results than the mainstream baselines in both training time and grading errors. As for future work, we would like to continue exploiting the potential of rubric, such as building multiple simple classifiers to classify whether each condition is satisfied, or try to classify more subtle type of answers. In addition, we would like to use other publicly available datasets with rubric tables in the future to test the effectiveness of our approach.

Acknowledgment

This work was supported in part by Grant-in-Aid for Scientific Research proposal numbers (JP21H00907, JP21K11847, JP20H01728, JP20H04300, JP19KK0257). We would like to express our deepest gratitude to them.

References

[1] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv:1708.00055*, 2017.

[2] Li-Hsin Chang, Iiro Rastas, Sampo Pyysalo, and Filip Ginter. Deep learning for sentence clustering in essay grading support. *arXiv:2104.11556*, 2021.

[3] Aubrey Condor, Max Litster, and Zachary Pardos. Automatic short answer grading with sbert on out-of-sample questions. *International Educational Data Mining Society*, 2021.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018.

[5] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. *arXiv:1907.12009*, 2019.

[6] Mike Hardy. Toward educator-focused automated scoring systems for reading and writing. *arXiv:2112.11973*, 2021.

[7] Kato Hiroyuki, Ishioka Tsunenori, and Mine Tsunenori. Automatic short answer scoring using thesaurus-based data augmentation. *IEICE Tech. Rep.*, 2021.

[8] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. *arXiv:2011.05864*, 2020.

[9] Jiaqi Lun, Jia Zhu, Yong Tang, and Min Yang. Multiple data augmentation strategies for improving performance on automatic short answer scoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[10] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. A sick cure for the evaluation of compositional distributional semantic models. In *Lrec*, 2014.

[11] Ifeanyi G. Ndukwe, Chukwudi E. Amadi, Larian M. Nkomo, and Ben K. Daniel. Automatic grading system using sentence-bert network. In *Artificial Intelligence in Education*, 2020.

[12] Rian Adam Rajagede. Improving automatic essay scoring for indonesian language using simpler model and richer feature. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 2021.

[13] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv:1908.10084*, 2019.
[14] V Sreevidhya and Jayasree Narayanan. Short descriptive answer evaluation using word-embedding techniques. In *2021 12th International Conference on Computing Communication and Networking Technologies*, 2021.
[15] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *arXiv:2103.15316*, 2021.
[16] Chul Sung, Tejas Dhamecha, Swarnadeep Saha, Tengfei Ma, Vinay Reddy, and Rishi Arora. Pre-training bert on domain resources for short answer grading. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019.
[17] Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*, 2019.
[18] Zichao Wang, Andrew S Lan, Andrew E Waters, Phillip Grimaldi, and Richard G Baraniuk. A meta-learning augmented bidirectional transformer model for automatic short answer grading. In *EDM*, 2019.