# Incorporating Question Information to Enhance the Performance of Automatic Short Answer Grading

Shuang Chen and Li Li(✉)

School of Computer and Information Science, Southwest University,
Chongqing, China
chen60423351@email.swu.edu.cn, lily@swu.edu.cn

**Abstract.** Automatic short answer grading (ASAG) is focusing on tackling the problem of automatically assessing students' constructed responses to open-ended questions. ASAG is still far from being a reality in NLP. Previous work mainly concentrates on exploiting feature extraction from the textual information between the student answer and the model answer. A grade will be assigned to the student based on the similarity of his/her answers and the model answer. However, ASAG models trained by the same type of features lack the capacity to deal with a diversity of conceptual representations in students' responses. To capture multiple types of features, prior knowledge is utilized in our work to enrich the obtained features. The whole model is based on the Transformer. More specifically, a novel training approach is proposed. Forward propagation is added in the training step randomly to exploit the textual information between the provided questions and student answers in a training step. A feature fusion layer followed by an output layer is introduced accordingly for fine-tuning purposes. We evaluate the proposed model on two datasets (the University of North Texas dataset and student response analysis (SRA) dataset). A comparison is conducted on the ASAG task between the proposed model and the baselines. The performance results show that our model is superior to the recent state-of-the-art models.

**Keywords:** Automatic short answer grading · Natural language processing · Classification

## 1 Introduction

With the rapid development of online education, there are many challenges for educators. One challenge is for instructors scoring work to assess the gained knowledge from students and to be able to provide instructive feedback to teachers. However, manual grading is tedious and has a low degree of reproducibility. The computer-aided assessment has been facilitated in schools and colleges for several years currently, but primarily for the objective test with strained answers such as Multiple Choice Questions. A previous study did specify which nature

H. Qiu et al. (Eds.): KSEM 2021, LNAI 12817, pp. 124–136, 2021.
https://doi.org/10.1007/978-3-030-82153-1_11

of objective tests was deficient capture multiple aspects of acquired knowledge from the student. Such as reasoning and comprehension [1].

Thus, assessment of some form of a free response by students on open-ended questions could be a major focus of current research. Specifically, We are interested in fill-in-the-gap and essay-style short answers that are between a few words and a few sentences long [2,3]. Previously Automatic Short-Answer Grading (ASAG) is the procedure of assigning grades to student provided free-text answers either by comparing it with the corresponding model answers or pattern-based answers extracted from student answer [4,5]. Grading student-constructed answers could be a complicated natural language task attributed to linguistic diversity (the same answer can be phrased in numerous ways). Therefore ASAG is an important research area.

The methods of calculating the semantic similarity of two texts have been well researched in Natural Language Processing(NLP) literature [6]. These works mainly based on supervised learning technology are all around the text-similarity (synonymously, overlap, correspondence, entailment etc.) in NLP [7,8]. The ASAG system's general procedure is that features are extracted from model answers and student answers that have been graded by the instructor or grader and fed these features into various classification or regression models for training. The trained model will automatically mark raw student answers fraction. ASAG systems can be regarded in two ways:

**Classification/labeling task** (Table 1, the 2rd column): The student answer will be assigned to one of a set of categories i.e., 'correct', 'partially correct incomplete', 'contradictory', 'irrelevant', 'non domain'.

**Regression task** (Table 1, the 3rd column): Assign a mark/grade to the student answer based on the similarity with the corresponding model answer.

**Table 1.** Sample in the middle are from the ScientsBank [9] subset of student response analysis(SRA dataset) corpus, and the 3rd column are from the undergraduate Data Structure course(CS dataset) [8], the scores or label of each student's answer are manually assigned.

| | | |
|---|---|---|
| Question | Throwing a ball uses a hinge joint and a ball-and-socket joint. Describe how each of these 2 joints moves when you throw a ball. The hinge joint — | What is the role of a prototype program in problem solving? |
| Reference answer | Moves back-and-forth | To simulate the behaviour of portions of the desired software product |
| Student answer-1 | Up and down if it did not go up and down you could not throw a ball. (correct) | You can break the whole program into prototype programs to simulate parts of the final program. (5.0/5.0) |
| Student answer-2 | The socket joint makes sure that you do not break your wrist. (irrelevant) | A prototype program is a part of the Specification phase of Software Problem Solving. It is employed to illustrate how the key problem or problems will be solved in a program, and sometimes serves as a base program to expand upon. (4.5/5.0) |

However, the previous ASAG model had remarkable shortcomings. With the diversity of the conceptual representation in the context, student-constructed responses are diverse and sophisticated. A correct answer has relevance to the question and the reference answer. Meanwhile, students' responses may be different from the reference answer but still correct, they may be similar to the reference answer but incorrect. For example, in Table 1, both questions have a brief reference answer. Each answer is different from the reference answer but is correct, and interestingly is highly similar to the question. We analyze that the feature extracted from the answer and the reference answer pair does not effectively address the variety of conceptual representations. We can use the method of adding prior knowledge to capture multiple types of features to enhance performance. Here, we want to exploit the textual information of the questions and student answers to enable the ASAG system to mine meaningful semantic features that the previous model could not hit.

In this paper, we propose a novel ASAG system based on the Transformer [10] network. We proposed a novel Feature Fusion layer based on the pooling layer that was deployed as fine-tuning. Then, we design a novel train approach by adding a forward propagation on a random training step, which takes the string of the reference answer and answers into the model to do forward propagation to get the first output. Consecutively, we input the string of the question and student answer into the same model to do another forward propagation to get the second output. Finally, feed the two outputs obtained above into the Feature Fusion layer filtering out multiple facets of semantic features, followed by a flexible output layer for generating a score/label. Our contribution is as follows

– We propose a novel training approach to incorporate textual information of questions and answers. The method has the capacity to enable the ASAG system to capture multiple aspects of semantic features.
– We customize an ASAG system for the proposed training method comprising a novel Feature Fusion layer based on the pooling layer over the Transformer-Encoder network, followed by a flexible output layer for generating a score/label.
– Experiments were evaluated on two publicly available datasets. We compare the performance of several popular Transformer models on the ASAG task. Extensive experimental results illustrate the effectiveness of our model and outperform the previous ASAG model in most metrics.

## 2 Related Work

Numerous approaches have been proposed for the grading of short answers. The Oxford-UCLES system [4] requires manually crafted patterns by using a set of keywords and synonyms to search for a new pattern through a text window. C-Rater [11] generates a word set that is extracted from the model answer set, and then the corresponding student's answer is matched with this word set for scoring. Many text similarity methods have been considered, and the assigned scores are measured relying on the correlation between the student and model answers, using text similarity measures such as knowledge-based, corpus-based [7,8,12]and word embedding [13].

To improve the performance of the model, many researchers have tried to combine ensemble learning. Divide the feature set into several feature subsets to train the classifier, and ensemble the different classifiers trained [14]. Similarly, there is the ensemble of different regression models with the same feature set [13]. The ASAG system, which combines domain adaptive techniques and ensemble two classifiers, is designed to grade students [15]. The above work requires feature engineering. There is also a deep learning model in the ASAG system. Which contains three neural network blocks: Siamese bidirectional LSTMs(bi-LSTMs), EMD(Earth mover's distance) pooling layer, and regression layer. It can assign a score to student answers by optimizing above mentioned neural network blocks in order [16]. However, ASAG similar to other nontrivial NLP tasks that have limited training data.

The NLP community has recently proposed many general pre-trained language models, which can be transferred and fine-tuned seamlessly for any downstream tasks. Bidirectional Encoder Representations from Transformers(BERT [17]) has been proven to achieve state-of-the-art results through fine-tuning in a large number of tasks. It is a trained deep language model that can simultaneously combine left and right context information on all layers. By training the BERT model from corpus resources in a specific domain, fine-tuning it in the ASAG task will achieve superior performance [18].

However, all of these ASAG systems leverage the same type of features extracted from textual information between the reference answer and the student answer. The results in the ASAG system lack of capability to tackle the diversity of conceptual representations in response. More specifically, these ASAG systems are incapable to discriminate against the ground truth for student answers with low or no similarity to the reference answer. In this article, we focus on methods to add prior knowledge through textual information of the question that was discarded by most previous researchers to capture multiple aspects of semantic features.

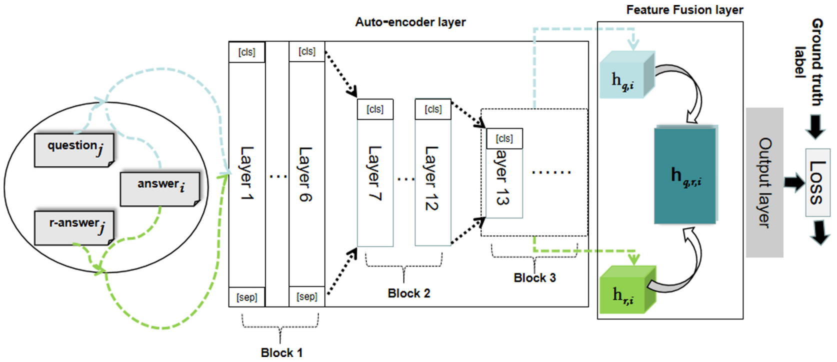

**Fig. 1.** Schematic of ASAG system.

## 3 Our System

The black circle on the left of Fig. 1 shows input data for the system. $question_{\mathrm{j}}$, $r - answer_{\mathrm{j}}$ and $answer_{\mathrm{i}}$ represents the case of the input data. $question_{\mathrm{j}}$ represents the jth question in the dataset, $r - answer_{\mathrm{j}}$ represents the $question_{\mathrm{j}}$ corresponding reference answer, $answer_{\mathrm{i}}$ denotes the ith responses to jth question. Auto-encoder Layer is based on the Funnel-Transformer-Encoder model, with three blocks and six layers in each block. We propose a novel approach to training a model by dividing each training step into two stages. The blue dotted line represents the first stage fed the sequence of $question_{\mathrm{j}}$ and $answer_{\mathrm{i}}$ into the Auto-encoder Layer do forward propagation. Immediately, the green dotted line represents the second stage fed the sequence of r$-answer_{\mathrm{j}}$ and $answer_{\mathrm{i}}$ into the same Encoder Layer do another forward propagation. The output of two forward propagations are fed into our proposed Feature Fusion layer based pooling layer to filter out meaningful semantic features blue cube $h_{\mathrm{q,i}}$, green cube $h_{\mathrm{r,i}}$. Further fusing semantic features $h_{\mathrm{q,i}}$ and $h_{\mathrm{r,i}}$ in Feature Fusion Layer to obtain the dark blue square $h_{\mathrm{q,r,i}}$, which is the capture multiple aspects of the semantic feature. Then drop $h_{\mathrm{q,r,i}}$ into the output layer and the assessment score of the ith response by the ASAG system.

### 3.1 Auto-Encoder Layer

Funnel-Transformer [19] is proposed to compress the hidden layer computation reduction of the whole sequence, and the structure is similar to the Transformer. But the difference is made up of multiple layers of blocks stacked and the sequence length of each block is gradually reduced through a pooling as follow:

$$\mathbf{h}^{'} \leftarrow Pooling(\mathbf{h}) \tag{1}$$

After a pooling, the sequence length of $\mathbf{h}^{'}$ will be less than the sequence length of $\mathbf{h}$, which is $L^{'} < L$. The new multi-head self-attention can be expressed as

$$\mathbf{h} \leftarrow LayerNorm(\mathbf{h}^{'} + S - Attn(Q = \mathbf{h}^{'}, KV = \mathbf{h})) \tag{2}$$

Where h is still used as a role of key and value vector, which is to reduce the loss of information after passing through the pooling. In this article, the Funnel-Transformer-encoder will be used as the Auto-encoder of the proposed system.

### 3.2 Feature Fusion Layer

In this article, We propose a novel approach to divide each training step into a two-stage. More specifically, the first stage is to input the answer and question pairs in the form of "[CLS] answer [SEP] question [SEP]" into the Auto-encoder layer to do forward propagation to obtain the output of sequence representation as Eq. (3). Similarly, in the second stage, to input the answer and model answer pairs in the form of "[CLS] answer [SEP] model answer [SEP]" into the same Auto-encoder layer does a second forward propagation to obtain the sequence representation as Eq. (4). Due to Funnel-Transformer-encoder, the length of the sequence decreases as the number of blocks increases. Then the sequence length of all layers in the last block is greatly reduced. Avoid waste the sequence representation, distinct from takes the hidden state of the first position token ([cls] token) of the sequence as aggregate representation, we can filter the sequence representation as aggregate representation. Therefore, we propose a novel Feature Fusion layer based on the pooling layer and put the output of two forward propagations into the Feature Fusion layer, which can be described as.

$$h^{''}_{\mathfrak{q},\mathfrak{i}} \leftarrow LayerNorm(h^{'}_{\mathfrak{q},\mathfrak{i}} + S - Attn(Q = h^{'}_{\mathfrak{q},\mathfrak{i}}, KV = h_{\mathfrak{q},\mathfrak{i}})) \tag{3}$$

$$h^{''}_{\mathfrak{r},\mathfrak{i}} \leftarrow LayerNorm(h^{'}_{\mathfrak{r},\mathfrak{i}} + S - Attn(Q = h^{'}_{\mathfrak{r},\mathfrak{i}}, KV = h_{\mathfrak{r},\mathfrak{i}})) \tag{4}$$

$$h^{'''}_{\mathfrak{q},\mathfrak{i}} \leftarrow Pooling(h^{''}_{\mathfrak{q},\mathfrak{i}}) \tag{5}$$

$$h^{'''}_{\mathfrak{r},\mathfrak{i}} \leftarrow Pooling(h^{''}_{\mathfrak{r},\mathfrak{i}}) \tag{6}$$

where $h^{''}_{\mathfrak{q},\mathfrak{i}}$ is sequence representation of questions and corresponding student responses. $h^{'''}_{\mathfrak{q},\mathfrak{i}}$ is meaningful semantic features filtered out from textual information of questions and answers. $h^{''}_{\mathfrak{r},\mathfrak{i}}$ and $h^{'''}_{\mathfrak{r},\mathfrak{i}}$ is also obtained in the above manner. The difference is $h^{'''}_{\mathfrak{r},\mathfrak{i}}$ captured another type of semantic feature. Further we can fuse both $h^{'''}_{\mathfrak{q},\mathfrak{i}}$ and $h^{'''}_{\mathfrak{r},\mathfrak{i}}$ semantic features aim to capture multiple aspects of semantic feature, which can be described as.

$$\mathbf{h}^{''} \leftarrow Pooling([h^{1}_{\mathfrak{q},\mathfrak{r},\mathfrak{i}} : \cdots : h^{\mathfrak{j}}_{\mathfrak{q},\mathfrak{r},\mathfrak{i}}]) \tag{7}$$

where $h^{\mathfrak{j}}_{\mathfrak{q},\mathfrak{r},\mathfrak{i}} = [h^{'''}_{\mathfrak{q},\mathfrak{i}} : h^{'''}_{\mathfrak{r},\mathfrak{i}}]$ is expressed as a matrix of $h^{'''}_{\mathfrak{q},\mathfrak{i}}$ and $h^{'''}_{\mathfrak{r},\mathfrak{i}}$ of the j-th ($\mathfrak{j} \in num\_layer$) layer concatenated according to feature dimensions. To better filter out meaningful semantic features, $h^{1}_{\mathfrak{q},\mathfrak{r},\mathfrak{i}}, \cdots, h^{\mathfrak{j}}_{\mathfrak{q},\mathfrak{r},\mathfrak{i}}$means that output of each layer of the last block in the Auto-encoder layer, concatenated according to the sequence dimension. Finally it is filtered the sequence dimension to obtain $\mathbf{h}^{''}$.

### 3.3 Output Layer

Here, we have captured multiple aspects of semantic feature $\mathbf{h}^{''}$ . Further, the following.

$$S = softmax(tanh(\mathbf{h}^{''}W_0)W_1) \tag{8}$$

where $W_0$ and $\mathbf{h}^{''}$ have the same dimensions. After nonlinear mapping is performed through tanh function, linear matrix transformation is performed through $W_1$. For classification tasks and fed to the softmax classifier. Although the output after the Compress layer is a set of values and the number of dimensions and label categories are the same, it did not directly correspond to answer labels. It needs to be fed into a softmax classifier. The loss function on all labeled data is defined as cross-entropy error:

$$\mathcal{L}_{cls} = -\sum_{\mathrm{i} \in T_I}\sum_{f=0}^{F} T_{\mathrm{i}f} ln S_{\mathrm{i}f} \tag{9}$$

where $T_I$ is the set of graded student answers, F is the dimension of the output features, and $T_{\mathrm{i}f}$ is the ground truth of the ith answer in $T_I$.

For regression tasks, remove the softmax classifier and the output is a single value, but it will not directly match the answer score. The final loss function can be the L2 (mean square) error between the above output prediction and ground truth. The overall ASAG system is schematically illustrated in Fig. 1.

## 4 Experiments

### 4.1 Dataset

We use two publicly available datasets for evaluation, shwon as follows:

**CS dataset**[1] [8]**:** This dataset is provided by the two examinations of the Data Structure course of a class of undergraduates at the University of North Texas. It contains 2442 student answers and 87 questions and the corresponding model answers are spread across ten assignments. The answer has been scored from 0 to 5 independently by two human graders. Their average score will be used as the gold standard, The Inter Annotator Agreement(IAA) between the two annotators Pearson Correlation Coefficient = 0.586.

**SRA dataset**[2]**:** This dataset is part of the“Student Response Analysis” (SRA) in the Semantic Evaluation (SemEval) workshop in 2013 [20], it contains two types of datasets:

**- ScientsBank:** This dataset has about 10,000 student answers to around 197 questions, which belonging to 15 different scientific domains. The answers have been graded as explained in [9].

**- Beetle:** This dataset has about 3000 answers and 56 questions in Basic Electricity and Electronics domains extracted from the interactions with the

[1] http://web.eecs.umich.edu/~mihalcea/downloads/ShortAnswerGrading_v2.0.zip.

[2] https://www.cs.york.ac.uk/semeval-2013/task7/index.php%3Fid=data.html

Beetle-II Tutorial Dialogue system [21]. The student answers can refer to single or multiple reference answers. Each reference answer has a category from ‘BEST’, ‘GOOD’, ‘MINIMAL’ or ‘KEYWORD’.

All student answers contained in the SRA dataset [20] are labeled by manual graders, and categories are in any of the five categories named, Correct, Partially Correct Incomplete, Contradictory, Irrelevant, Non-domain. In case of ScientsBank corpus, it contains three types of test sets [9]: Unseen-answers (UA), Unseen-questions (UQ) and Unseen-domains (UD). However, for Beetle corpus, only two types of test sets are included: Unseen-answers (UA) and Unseen-questions (UQ). All the above five-way datasets are considered for optimization.

### 4.2 Experiment Design

**Settings.** Our system will evaluate on classification tasks and regression tasks, so we have two sets of experimental settings. For classification tasks on the SRA dataset, experiment and debug other parameters, set the learning rate to 3e-5, the drop out to 0.1, and the epoch to 6. The cross-entropy loss function sets the parameter weight, which is set to (5., 2., 1., 2., 1.) for scientbank’s UQ test set, and (1., 1., 1., 1., 15.) for the rest of the test sets. For the regression task on the CS dataset, we randomly divide 10% of the dataset used for testing, and the remaining dataset (about 2197 student answers) for training (in 12-fold cross-validation), the learning rate is set to 4e−5, the rest of the settings are a similar as the classified tasks. Besides, we use the AdamW [22] optimizer, with a linear learning rate schedule with 20% of train steps to warm-up. The padding size of the two experiments is 90.

**Evaluation Metrics.** The results reported using evaluation metrics Weighted-average F1 score and Macro-average F1 score consider all categories for the classification task. In the regression task, there are two types of evaluation metrics for the ordinal labels predicted by ASAG. RMSE is used here as a metric of value deviation, and Pearson’s is the most popular correlation coefficient.

**Table 2.** We make tests on ScientBank, Beetle and CS dataset. Where UA is denoted as Unseen-answers test set, UQ is denoted as Unseen-questions test set and UD is denoted as Unseen-domains test set.

| Model | Scientbank | | | | | | Beetle | | | | CS dataset | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Wighted F1 | | | Macro F1 | | | Wighted F1 | | Macro F1 | | Pearson | RMSE |
| | UA | UQ | UD | UA | UQ | UD | UA | UQ | UA | UQ | | |
| BERT-question | 0.541 | 0.365 | 0.394 | 0.455 | 0.251 | 0.375 | 0.687 | 0.579 | 0.626 | 0.557 | 0.621 | 0.880 |
| BERT-reference | 0.644 | **0.556** | 0.526 | 0.480 | **0.398** | 0.510 | 0.784 | **0.702** | 0.704 | 0.615 | 0.722 | 0.778 |
| Our approach | **0.659** | 0.541 | **0.534** | **0.498** | 0.388 | **0.512** | **0.790** | 0.693 | **0.713** | **0.660** | **0.754** | **0.736** |

### 4.3 Proposed Training Approach

In this experimental part, we only use BERT [10] as the encoder layer in our ASAG system, because we want to verify that the idea of incorporating question information is feasible. As shown in Table 2, BERT-reference indicates that the input is the reference answer and answer pair, BERT-question is also the same meaning. The result is our proposed approach has better results. We analyze that the proposed method can capture multiple aspects of semantic features compared to other systems, and the model can learn a better feature distribution. The following experiments will all adopt this training approach.

**Table 3.** Influence of encoder and ablation experiment table

| | Model | Scientbank | | | | | | Beetle | | | | CS dataset | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Wighted F1 | | | Macro F1 | | | Wighted F1 | | Macro F1 | | Pearson | RMSE |
| | | UA | UQ | UD | UA | UQ | UD | UA | UQ | UA | UQ | | |
| A | BERT-base | 0.659 | 0.541 | 0.534 | 0.498 | 0.388 | 0.512 | 0.790 | 0.693 | 0.713 | 0.660 | 0.754 | 0.736 |
| | ROBERTA-base | 0.678 | 0.572 | 0.560 | 0.625 | 0.407 | 0.538 | **0.794** | 0.722 | 0.720 | **0.685** | 0.767 | 0.735 |
| | ELECTRA-base | 0.706 | **0.603** | 0.583 | 0.549 | 0.438 | 0.597 | 0.764 | 0.720 | 0.637 | 0.616 | 0.754 | 0.741 |
| | Funnel-B6-6-6 | **0.717** | 0.600 | **0.602** | **0.667** | **0.582** | **0.630** | 0.789 | **0.735** | **0.768** | 0.622 | **0.796** | **0.678** |
| | Funnel-B4-4-4 | 0.689 | 0.597 | 0.562 | 0.587 | 0.434 | 0.575 | 0.779 | 0.697 | 0.728 | 0.616 | 0.795 | 0.683 |
| | Funnel-B6-3*2-3*2 | 0.655 | 0.585 | 0.565 | 0.488 | 0.398 | 0.560 | 0.775 | 0.714 | 0.688 | 0.606 | 0.782 | 0.698 |
| B | Funnel-layers(−6) | 0.717 | **0.600** | 0.602 | **0.662** | **0.582** | **0.630** | **0.789** | **0.735** | **0.768** | **0.622** | **0.796** | **0.678** |
| | Funnel-layers (−5) | **0.724** | 0.575 | 0.593 | 0.562 | 0.513 | 0.589 | 0.779 | 0.707 | 0.692 | 0.602 | 0.747 | 0.760 |
| | Funnel-layers (−4) | 0.708 | 0.589 | 0.571 | 0.609 | 0.529 | 0.590 | 0.780 | 0.729 | 0.689 | 0.618 | 0.789 | 0.690 |
| | Funnel-layers (−3) | 0.684 | 0.572 | 0.574 | 0.513 | 0.516 | 0.558 | 0.768 | 0.687 | 0.710 | 0.576 | 0.751 | 0.745 |
| | Funnel-layers (−2) | 0.703 | 0.510 | **0.604** | 0.543 | 0.449 | 0.629 | 0.767 | 0.725 | 0.698 | 0.614 | 0.761 | 0.733 |
| | Funnel-layers (−1) | 0.707 | 0.552 | 0.596 | 0.609 | 0.501 | 0.612 | 0.787 | 0.684 | 0.728 | 0.606 | 0.760 | 0.731 |
| C | Funnel-Avg_pooling | **0.717** | **0.600** | **0.602** | **0.662** | **0.582** | **0.630** | **0.789** | **0.735** | **0.768** | **0.622** | **0.796** | **0.678** |
| | Funnel-Max_pooling | 0.676 | 0.479 | 0.561 | 0.502 | 0.414 | 0.544 | 0.769 | 0.712 | 0.715 | 0.609 | 0.768 | 0.743 |
| | Funnel-Dropping | 0.707 | 0.310 | 0.596 | 0.609 | 0.260 | 0.612 | 0.787 | 0.684 | 0.728 | 0.606 | 0.775 | 0.710 |
| D | Funnel-Normal_set | – | – | – | – | – | – | **0.789** | **0.735** | **0.768** | 0.622 | – | – |
| | Funnel-Optimal_set | – | – | – | – | – | – | 0.772 | 0.717 | 0.685 | 0.664 | – | – |
| | Funnel-GOOD_set | – | – | – | – | – | – | 0.773 | 0.659 | 0.675 | 0.563 | – | – |
| | Funnel-BEST_set | – | – | – | – | – | – | 0.755 | 0.676 | 0.657 | **0.668** | – | – |

### 4.4 Influence of Encoder

In this section, we selected the few most popular transformers to replace Funnel-Transformer for comparison experiments. It is BERT-uncased-base, ROBERTA [23]-base, ELECTRA [24]-base-discriminator, Funnel-B6-6-6, Funnel-B4-4-4 and Funnel-B6-3*2-3*2. The above six pre-trained models are all from huggingface[3]. Except for Funnel-B6-6-6 (6 represent each six layers as a block) and Funnel-B6-3*2-3*2 has 18 layers, each other pre-trained model has 12 layers, every layer has 12 heads, and the hidden layer size is 768 dimensions.

The results of Block A in Table 3 show that Funnel-B6-6-6 has the best performance compared to other models. We believe that the number of layers of

[3] https://huggingface.co/.

Funnel-B6-6-6 is the key point. The reason is sequence length becomes shorter after a pooling operation, which leads to savings memory, further can increase the number of layers of stacked transformers from 12 to 18 and boost the model capacity. However, Funnel-B6-3*2-3*2 also has the same number of stacked layers and the difference is that every two layers share parameters in the 2nd and 3rd blocks. The reason for the loss performance is the number of parameters will be less than Funnel-B6-6-6. Compared with the model mentioned above, Funnel-B6-6-6 can achieve better results at the expense of more computing power.

### 4.5 Ablation Experiment

We compare some important points about our system in this section.

**Encoder Layer Comparative:** In this section, we want to know which aggregate representation of layers is worth utilizing(can be regarded as the multi_layer of Sect. 3 and formula 7). According to [25], Attention heads in the last few layers of BERT in medium metastable states [25], and still have learning ability after pre-trained. Therefore, we only compare the last six layers of the encoder for experimental validation. The result is shown in block B in Table 3. We believe that splicing the aggregate representation of the last six layers (expressed as layers($-6$)) and the best result can be achieved. In the following ablation experiments, we adopt this fixed setting.

**Feature Fusion Layer Comparative:** Our Feature Fusion layer uses the Average pooling operations (as shown in block C in Table 3). For simplicity, we only experiment with stride 6 and window size 6 in this work. We compared the results after replacing it with another Max pooling operation and the ASAG system after dropping Feature Fusion layer (denoted as Funnel-Dropping). Through the results, we believe that compared to Average pooling, Max pooling operation will lose more valuable feature information during the compression sequence.

**Beetle Dataset Analysis:** We use all the reference answers corresponding to the question as a Normal set, the previous experiments also used the Normal set. Then we manually filter out the model answers, we put the categories of each conference answers are 'BEST' and 'GOOD' as BEST set and GOOD set, respectively, and further combine the above two sets as an Optimal set. We have done four comparative experiments and aim to analyze whether the category of the reference answer will have a significant impact on the result. As shown block D in Table 3, from top to bottom, indicates that used reference answer set in the experiment. Our analysis that the category of the reference answer will not lead to improve the result, but the number of reference answers will increase the result. This is because the more prior knowledge the model can mine meaningful semantic features.

### 4.6 Comparison with State-of-the-Art Systems

Our proposed ASAG system will be compared with other state-of-art ASAG systems. The ASAG systems that have been considered are Mohler [8], Earth

Mover's Distance-based ASAG System [16], Iterative Ensemble [15], ETS [14] and Feature Engineering and Ensemble-Based [13]. Table 4 shows that the comparison between our proposed system and the above systems includes regression tasks and classification tasks.

**Table 4.** Our proposed system is compared with the recently out-of-the-art ASAG system

| ASAG systems | Scientbank | | | | | | Beetle | | | | CS dataset | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Wighted F1 | | | Macro F1 | | | Wighted F1 | | Macro F1 | | Pearson | RMSE |
| | UA | UQ | UD | UA | UQ | UD | UA | UQ | UA | UQ | | |
| Mohler [8] | – | – | – | – | – | – | – | – | – | – | 0.518 | 0.978 |
| Earth Mover's Distancebased [16] | – | – | – | – | – | – | – | – | – | – | 0.649 | 0.830 |
| ETS [14] | 0.625 | 0.356 | 0.434 | 0.581 | 0.274 | 0.339 | 0.705 | 0.614 | 0.619 | 0.552 | – | – |
| Iterative Ensemble [15] | 0.672 | 0.518 | 0.507 | 0.612 | 0.415 | 0.402 | – | – | – | – | – | – |
| Feature Engineering and Ensemble-Based [13] | **0.925** | **0.658** | **0.656** | **0.899** | 0.527 | 0.505 | 0.7091 | 0.6248 | 0.5969 | 0.5923 | 0.703 | 0.793 |
| Our System | 0.717 | 0.6004 | 0.6019 | 0.6671 | **0.5823** | **0.6298** | **0.7889** | **0.7348** | **0.7683** | **0.6219** | **0.7961** | **0.6776** |

The results show that the performance of the proposed system exceeds the most recent ASAG systems. Compared with [13], our system has highly improved performance both in the case of regression tasks and for classification tasks in Beetle. But for the classification task on ScientsBank, we only have a huge improvement on the Macro F1 metric of UQ and UD compared to [13], and the Weighted F1 and Macro F1 of UA have a significant improvement over [16] but lower than [13]. We analyze that our proposed system perform poorly on long-tailed dataset such as ScientsBank. More specifically, the number of students answers in category Non_doamin is minimal just 23 compared to 4969 students answers, which makes it difficult for our proposed neural network based system to learn the feature distribution of samples. Therefore, the performance of the proposed system on classification tasks for the ScientBank dataset will underperformance compared to the beetle. We analyze that feature engineering in [13] tackle long-tailed dataset is helpful, which it can extract a variety of text similarity features. In summary, our model achieved the best grading performance on the regression task with Pearson = 0.796 and RMSE = 0.678. The best labeling performance on the classification task in case of Beetle Dataset with Weighted F1-score = 0.789, Macro F1-score = 0.768 (UA test set) and Weighted F1-score = 0.735, Macro F1-score = 0.622 (UQ test set).

## 5 Conclusion

In this work, we tested with several popular Transformers on the ASAG task, and the Funnel-Transformer has significant results compared to other models. We propose a novel ASAG system comprising a novel Feature Fusion layer based on the pooling layer over the Transformer-Encoder network. Further, our proposed novel training approach incorporates question information to obtain more prior knowledge. Our proposed ASAG system can effectively tackle the diversity of conceptual representations in a student response. Extensive experiments demonstrate the superior performance on the two publicly available datasets and surpass the most recent out-of-the-art ASAG systems.

**Acknowledgement.** This research was supported by NSFC (Grants No. 61877051). Li Li is the corresponding author for the paper.

## References

1. Conole, G., Warburton, B.: A review of computer-assisted assessment. Res. Learn. Technol. **13**(1), 17–31 (2005)
2. Burrows, S., Gurevych, I., Stein, B.: The eras and trends of automatic short answer grading. Int. J. Artif. Intell. Educ. **25**(1), 60–117(2015)
3. Roy, S., Narahari, Y., Deshmukh, O.D.: A perspective on computer assisted assessment techniques for short free-text answers. In: Ras, E., Joosten-ten Brinke, D. (eds.) CAA 2015. CCIS, vol. 571, pp. 96–109. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27704-2_10
4. Pulman, S.: Automarking 2: an update on the uclesoxford university research into using computational linguistics to score short free text responses (2004)
5. Thomas, P.: The evaluation of electronic marking of examinations. In: Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education, pp. 50–54 (2003). https://doi.org/10.1145/961511.961528
6. Pedersen, T., Patwardhan, S., WordNet, J.M.: Similarity - measuring the relatedness of concepts. In: AAAI, pp. 1024–1025. AAAI Press/The MIT Press (2004)
7. Mohler, M., Mihalcea, R.: Text-to-text semantic similarity for automatic short answer grading. In: EACL, pp. 567–575. The Association for Computer Linguistics (2009)
8. Mohler, M., Bunescu, R., Mihalcea, R.: Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: ACL, pp. 752–762. The Association for Computer Linguistics (2011)
9. Dzikovska, M.O., Nielsen, R.D., Brew, C.: Semeval-2013 task 7: the joint student response analysis and 8th recognizing textual entailment challenge. In: SemEval@NAACL-HLT, pp. 263–274. The Association for Computer Linguistics (2013)
10. Vaswani, A., Shazeer, N., Parmar, N.: Attention is all you need. In: NIPS, pp. 5998–6008 (2017)
11. Claudia, L., Martin, et al.: Automated scoring of short-answer questions. Comput. Hum. **37**, 92–96 (2003)
12. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. Unt Scholarly Works **1**, 775–780 (2006)
13. Sahu, A., Bhowmick, P.K.: Feature engineering and ensemble-based approach for improving automatic short-answer grading performance. IEEE Trans. Learn. Technol. **13**(1), 77–90 (2020)
14. Heilman, M., Madnani, N.: ETS: domain adaptation and stacking for short answer scoring. In: SemEval@NAACL-HLT, pp. 275–279. The Association for Computer Linguistics (2013)

15. Roy, S., Bhatt, H.S., Narahari, Y.: An iterative transfer learning based ensemble technique for automatic short answer grading. CoRR, abs/1609.04909 (2016)
16. Kumar, S., Chakrabarti, S., Roy, S.: Earth mover's distance pooling over siamese LSTMs for automatic short answer grading. In: IJCAI, pp. 2046–2052. ijcai.org (2017)
17. Devlin, J., Chang, M.W., Lee, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1), pp. 4171–4186. Association for Computational Linguistics (2019)
18. Sung, C., Dhamecha, T., Saha, S.: Pre-training BERT on domain resources for short answer grading. In: EMNLP/IJCNLP (1), pp. 6070–6074. Association for Computational Linguistics (2019)
19. Dai, Z., Lai, G., Yang, Y.: Funnel-transformer: filtering out sequential redundancy for efficient language processing. CoRR, abs/2006.03236 (2020)
20. Dzikovska, M.O., Nielsen, R., Brew, C.: Towards effective tutorial feedback for explanation questions: a dataset and baselines. In: HLT-NAACL, pp. 200–210. The Association for Computational Linguistics (2012)
21. Dzikovska, M.O., Isard, A., Bell, P.: BEETLE II: an adaptable tutorial dialogue system. In: SIGDIAL Conference, pp. 338–340. The Association for Computer Linguistics (2011)
22. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam. CoRR, abs/1711.05101 (2017)
23. Liu, Y., Ott, M., Roberta, N.G.: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692 (2019)
24. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generators. In: ICLR. OpenReview.net (2020)
25. Ramsauer, H., Schäfl, B., Lehner, J.: Hopfield networks is all you need. CoRR, abs/2008.02217 (2020)