Balancing Human Efforts and Performance of Student Response Analyzer in Dialog-based Tutors

Tejas I. Dhamecha, Smit Marvaniya, Swarnadeep Saha, Renuka Sindhgatta, and Bikram Sengupta

IBM Research-India
{tidhamecha,smarvani,swarnads,renuka.sr,bsengupt}@in.ibm.com

Abstract. Accurately interpreting student responses is a critical requirement of dialog-based intelligent tutoring systems. The accuracy of supervised learning methods, used for interpreting or analyzing student responses, is strongly dependent on the availability of annotated training data. Collecting and grading student responses is tedious, time-consuming, and expensive. This work proposes an iterative data collection and grading approach. We show that data collection efforts can be significantly reduced by predicting question difficulty and by collecting answers from a focused set of students. Further, grading efforts can be reduced by filtering student answers that may not be helpful in training Student Response Analyzer (SRA). To ensure the quality of grades, we analyze the grader characteristics, and show improvement when a biased grader is removed. An experimental evaluation on a large scale dataset shows a reduction of up to 28% in the data collection cost, and up to 10% in grading cost while improving the response analysis macro-average F1.

Keywords: Student Response Analysis, Dialog based Tutor, Data Annotation and Collection Cost, Question Difficulty, Student Ability, Grader Agreement

1 Introduction

Student Response Analysis (SRA) [8] is the task of assessing and grading a student response in comparison to a reference answer for the given question. It is an integral component of any dialog-based Intelligent Tutoring System (ITS), specifically to enable Socratic tutoring [19]. Such ITS often asks focused questions prompting student to provide short answers. Therefore, the SRA task is equivalent to short-answer grading. SRA is often modeled as a classification problem. Obtaining an adequately labeled dataset to train classifiers for SRA is a critical step in building any dialog-based tutoring system. The collection of a dataset for this purpose involves creating questions, creating reference answers, collecting student answers, and having the responses graded by subject matter experts. This overall process is human-effort intensive, time consuming, and expensive. We observe that there is a scope for reducing and/or directing human-efforts, without reducing the classification performance.

Table 1: Characteristics of the large scale industry dataset.

Domain	Psychology
Number of questions	483
Number of students	1,164
Number of student responses	17,023
Number of words per response	Min: 1, Max: 198, Mean: 15.4, Std: 12.6
Number of graders	9
Number of grades per response	3
Number of grades	51,069
Grade categories	`CORRECT`, `PARTIAL`, `INCORRECT`, `NON-ANSWER`

In the process of building an SRA for an ITS in the psychology domain, we obtained a large scale industry dataset. The characteristics of this dataset are reported in Table 1. The dataset contains 17,023 answers collected from 1,164 students for 483 questions. Each student answer is graded by exactly 3 graders; with the categories of grade being `CORRECT`, `PARTIAL`, `INCORRECT`, and `NON-ANSWER`. Each student answered about 17 questions randomly selected by the system. A large portion ($\sim$70%) of students answered 13-20 questions. On an average, 34 answers were collected for each question.

Based on the analysis of the characteristics of data for SRA, this paper aims to present an iterative approach for efficient data collection. We experimentally validate the effectiveness of each stage of the proposed approach and show that significant human-effort can be reduced while improving classification performance.

The SRA is designed as a three-class (`CORRECT`, `PARTIAL`, and `INCORRECT`) classification task. For all our experiments, we choose majority vote of the three graders as the ground truth. The experiments are performed using a deep learning based sentence embedding approach, called InferSent [7][1] that reports state-of-the-art results on various Natural Language Understanding tasks. We generate the InferSent sentence embeddings for question (q), reference answer (r), and student answer (a), to obtain the feature representation ($|q-r|, |r-a|, |q-a|, q*r, r*a, q*a$) of a sample and then train a 3-class multinomial logistic regression. The next section discusses the related work and details our contributions.

2 Related Work

In the context of existing literature, we position our work along different aspects of obtaining relevant training data. These can be broadly classified into three categories - (i) reducing labeling effort, (ii) reducing labeling cost by using non-experts, and (iii) removing or rectifying noisy labels.

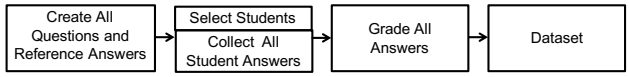
1. **Reducing labeling effort:** Semi-supervised learning techniques rely on clustering to select a subset of student answers for labeling [3,11,25]. Only the cluster centroids are labeled, and all the answers within the cluster are given the same label. Brooks et al. [6] proposed an efficient tool using a similar approach of forming clusters and labeling cluster centroids. They show

[1] `https://github.com/facebookresearch/InferSent`

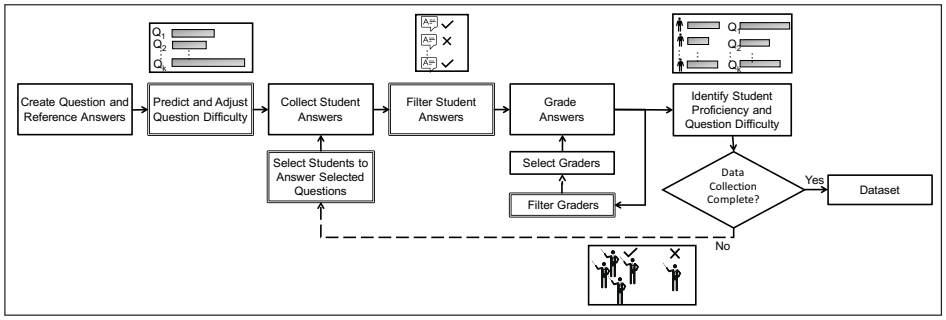
that the tool improves the teacher's efficiency by grading multiple students answers and providing feedback to them at once. As stated by the authors in one of the recent studies [25], clustering does not result in significant reduction of annotation effort if the responses are long, and is effective for very short student responses. Gweon et al.[10] proposed a two-stage approach where initially the text is automatically graded and then the grades are manually verified. Thus, the annotation task is transformed into a verification one, improving the efficiency of labeling efforts. Active learning (AL) based approaches [1,2,21] help in reducing the labeling efforts significantly by iteratively selecting a subset of samples to annotate. However, AL methods rely on the availability of large amount of unlabeled data and require the supervised model prior to data collection. AL based approaches need that the same classification pipeline is used during training data collection and for the final task. For many complex tasks, such as SRA, it is unlikely that the whole classification pipeline would be fixed beforehand due to its dependence on domain content. In other words, experiments with various choices of modules in the classification pipeline need to be performed; with the prerequisites of availability of data.

2. **Reducing labeling cost by using non-experts:** Instead of using the expert annotators only, one can employ a combination of expert and non-expert annotators to reduce the cost. The research in this direction involves reducing negative impacts of using non-experts while significantly lowering the cost. Snow et al. [23] proposed an approach to control the biases of non-expert annotators on Amazon Mechanical Turk crowdsourcing platform. In recent studies in crowdsourcing [14,15,16], various multi-stage annotation processes [4,15,16] are proposed to ensure the quality of annotation. The approaches in this category are limited to reducing the labeling cost and seldom focus on the content collection cost.
3. **Removing or rectifying noisy labels:** Research studies involving identification of noisy labels, either remove [9,20] or rectify [17,18,24] them. Techniques are proposed to improve the quality of labeled samples and the quality of the model built from labeled data via repeated labeling [22]. As the noisy labels can be, at times, attributed to certain annotators; Hsueh et al. [12] show that removing these annotators helps in improving the quality of the labels. A typical hypothesis behind this direction of research is that removal or rectification of noisy labels is critical to training a classifier. We also present an approach to identify noisy labels and a set of unlabeled data points to reduce labeling effort.

There is a sizable amount of research work pertaining to efficient and cost-effective labeling/annotation, in various domains of machine learning. In the context of dialog-based tutoring systems, much of the work has focused on reducing the annotation effort [3,11,25]. Our work considers the inter-dependence of various characteristics of the dataset and focuses on a comprehensive method of iteratively collecting, labeling, and refining data. The difficulty of the question and associated reference answer influences selection of relevant students to collect adequate student responses. Therefore, we believe that the literature per-



(a) Traditional approach.



(b) Proposed data collection approach.

Fig. 1: As opposed to the traditional monolithic data collection approaches, we propose a more involved iterative data collection approach with four intervention points: adjusting questions based on machine predicted difficulty, refining collected student answers, refining graders and identifying appropriate set of students and questions for successive iterations.

taining to efficient labeling is inadequate to make the overall human efforts cost effective for the problem presented.

To this end, this paper makes the following contributions to reduce human efforts for content creations and grading without affecting the classification performance.

- An iterative data collection approach for reducing human efforts pertaining to the content creation and grading.
- Automated approach to predict question difficulty.
- Approach for selecting questions and students iteratively to obtain a representative dataset.
- Approach for reducing the grading effort by filtering some of the student answers.

Further, we propose techniques to identify some of the content issues, pertaining to question difficulty, student proficiency, student answers, and grader bias. The next section details the iterative data collection approach.

3 Proposed Data Collection Approach

Each instance of training data for SRA consists of a question, a reference answer, a student identifier, a student answer, and a set of grades that different graders provide. As shown in Fig. 1a, the traditional procedure of creating training data involves following steps.

1. Questions (Q) and their reference answers (R) are created by subject matter experts.
2. Students (S) are chosen.

3. Answers (A) for the questions are collected.
4. Graders evaluate correctness of student answers and provide grades (G).

The i^{th} sample of the dataset, represented as $(q_i, r_i, a_i, s_i, g_i)$, consists of the question (q_i), reference answer (r_i), student identifier (s_i), student answer (a_i), and grades (g_i). The joint distribution for a generative model of M such samples can be mathematically expressed as following:

$$
\begin{aligned}
P(Q, R, S, A, G) &= \prod_{i=1}^{M} p(q_i, r_i, s_i, a_i, g_i) \\
&= \prod_{i=1}^{M} \underbrace{p(g_i|q_i, a_i, r_i, \theta^g)}_{\text{Grading}} \underbrace{p(a_i|q_i, s_i)}_{\substack{\text{Student} \\ \text{Answer}}} \underbrace{p(s_i|\theta^s)}_{\substack{\text{Student} \\ \text{Selection}}} \underbrace{p(r_i|q_i)p(q_i|\theta^q)}_{\substack{\text{Question and} \\ \text{Reference} \\ \text{Answer creation}}} \quad (1)
\end{aligned}
$$

where θ^g, θ^s, and θ^q are the parameters governing the characteristics of the graders, students, and questions ,respectively.

On analyzing the dependencies of the variables, it can be seen that if a question is poorly created, it can affect all the successive steps. Similarly, if a non-representative set of students is selected, the collected answers and therefore, the grades may provide limited information for training SRA. Ideally, the grades should be dependent on the question, reference answer, and student answer only; however, in real-world, the subjectivity of the graders also affect the grades.

To address these challenges, we propose a data collection approach as illustrated in Fig. 1b. It shows an iterative approach with intervention points for each of the four stages of data collection. The correspondence of these stages with components of generative models is shown in Eq. 1s. The boxes outlined with double lines in the figure denote these four stages where our proposed approach improves upon the existing methodology of data collection by minimizing the annotation cost and effort. The process starts at the question creation stage where we automatically predict the question difficulty. The next step involves collecting a subset of student answers consisting of $m < M$ responses (e.g. m = $\frac{M}{2}$). Further, some student answers are filtered out based on certain techniques. In the following step, the remaining student answers are graded. Based on graders' characteristics, some graders' annotations are discarded. Although this last stage does not yield any cost saving in the iteration, it is useful in maintaining quality of grades. Finally, using the tuple information of <Question, Student, Grade>, each student's proficiency and question's difficulty is estimated. These estimates are used for selecting students to answer specific questions in successive iterations to collect additional samples. This iterative approach thus enables focused answer collection activity, wherein students of certain proficiency are asked to provide answers for questions of certain difficulty only. This procedure helps in ensuring that the data set for training is balanced.

In the remaining part of this section, we quantify the overall cost involved in the data collection process and how our proposed approach is based on addressing the different cost components of data collection and annotation process.

Cost Estimation
Let n^q, n^a, and n^g be the number of questions, number of student answers collected per question, and number of grades obtained per student answer, respectively. Also, let c^q, c^s, and c^g be the costs of creating a question, collecting student answer for a question, and grading a student answer by a grader, respectively. The total cost is given by:

$$\text{Total Cost} = \underbrace{n^q c^q}_{\text{question creation}} + \underbrace{n^q n^a c^s}_{\text{answer collection}} + \underbrace{n^q n^a n^g c^g}_{\text{grading}} \quad (2)$$

If a question is poorly formed, the corresponding student answers as well as the grades are not useful. Similarly, if the students are selected poorly, the n^a responses may not provide the adequate spread of answer variations. Each of such student responses costs $c^s + n^g c^g$. If a grader is biased, it affects all the student answers graded by him/her, thus incurring cost in multiples of c^g.

By using the proposed approach in Fig. 1b, if we can filter out p questions in the difficulty prediction stage, followed by a reduction of m answers by focused student selection, and finally, remove k answers by filtering a selected set of answers, it will yield an overall cost reduction of $p\,(c^q + (k+m)\,(c^s + n^g c^g))$. The stage pertaining to filtering graders, may not result in cost saving as the graders are filtered only after the answers are graded. However, it is important to ensure the quality of grades. Additionally, based on graders performance, grader training may be directed in a more optimized way for future; which can be useful to organizations with large annotator workforce. The few next sections provide details and experimental evaluations of the claims pertaining to each stage of our approach on a large industry data set.

4 Predict and Adjust Question Difficulty

The first step of content creation is correctly formulating the questions. The questions should have carefully attuned difficulty levels. As an extreme case, having all the questions easy may not trigger complex thinking and, having all the questions difficult may discourage students from attempting. This part of question creation corresponds to $p(q_i|\theta^q)$ of the generative model in Eq. 1.

We want to predict question difficulty at the time of question creation, as this would ensure the desired mixture of question difficulty in the dataset. Although, the actual question difficulty is subjective and is relative to student proficiency, we observe that absolute question characteristics such as factoid/non-factoid, clarity of expression, and preciseness of the question play important role in defining its difficulty. Table 2 shows example questions with varying difficulties. These characteristics can be captured using natural language processing approaches. We propose a technique to learn the function f given by

$$f : question \rightarrow \beta \quad (3)$$

[1] For simplicity, we assume that the costs of question creation, answer collection and answer grading are uniform across questions and answers.

Table 2: Examples of question difficulty. As we move from easy to difficult question, there is a transition from objectivity to subjectivity, factoid to non-factoid, and preciseness to broadness.

Difficulty	**Question**
Easy	What are the two types of intelligence?
Medium	What is crystallized intelligence, and how does middle age impact it?
Difficult	How do older adults perform in intelligence activities?

where, β is the question difficulty [13]. This is one of the first attempts at predicting question difficulty prior to answer collection. The proposed technique involves a deep learning based feature extraction method followed by classification using multinomial Logistic Regression. Given a question, its embedding is computed using InferSent [7], a state-of-the-art deep learning framework for computing sentence embeddings. These question embeddings are subsequently used as features for the difficultly prediction task. InferSent is trained on the Stanford Natural Language Inference data and the sentence embeddings are computed using a bidirectional Long Short-Term Memory (LSTM) network over the pre-trained word embeddings. InferSent's pre-trained model has been shown to work well for computing embeddings in various NLP classification tasks. Thus, we also use InferSent to obtain a $4,096$ dimensional embedding for each question.

To obtain the true question difficulties β, Item Response Theory (IRT) [13] is used. It estimates a question's difficulty based on the grades of the student responses. In a broad sense, a question is considered difficult if it is likely to be answered correctly by high proficiency students. In this work, we used the Two-Parameter Logistic Model of IRT. The Two-Parameter Logistic Model [5] specifies the probability of a correct answer as a logistic distribution in which the items vary in terms of their difficulty and discrimination. It is typically applied to multiple choice or short response items that are scored either correct or incorrect, and do not allow for guessing.

While the question difficulty values (β) are more precise, the buckets or categories of difficulties are sufficient for assisting the question creation procedure. Therefore, we convert the problem into a prediction task where a given question is categorized as easy, medium, or difficult. β is uniformly divided into three intervals as shown below

$$\text{Difficulty Label} = \begin{cases} \text{Easy} & \text{if } -2 \leq \beta < -2/3 \\ \text{Medium} & \text{if } -2/3 \leq \beta < 2/3 \\ \text{Difficult} & \text{if } 2/3 \leq \beta \leq 2 \end{cases}$$

We train a 3-way Logistic Regression classifier and employ 5-fold cross-validation. Thus, for each sample, a difficulty label is predicted. Based on the predictions, the class-wise precision, recall, and F1 metrics are reported in Table 3. With our preliminary approach, we are able to identify easy and difficult questions with a reasonable recall of greater than 55%. This suggests that, to an extent, the proposed approach can assist humans in identifying question difficulties, thereby helping to ensure a desired mixture of difficulty levels. In a traditional approach, the content creators themselves can be asked to provide the difficulty levels.

Table 3: Classification performance of question difficulty prediction in terms of precision, recall, and F1.

	Precision	Recall	F1
Easy	49.12%	**55.63%**	52.17%
Medium	43.41%	35.90%	39.30%
Difficult	54.86%	**57.14%**	55.98%

However, it results in additional human-effort related cost. Additionally, we experimented with reference answers, independently and in conjunction with the question, to predict the difficulty. On this dataset, the reference answers did not improve the results. However, our proposed data collection approach does not rule out reference answers' utility.

5 Selecting Students for Answering Selected Question

Selection of students is an equally important aspect of the data collection process. The dataset may get biased towards correct class if the large portion of high proficiency students is selected. The classifier will thus end up not seeing enough samples of `PARTIAL` and `INCORRECT` classes, leading to a class imbalance problem. This will eventually lead to a very strict SRA. Therefore, it is important to select students carefully. The aspect of student selection corresponds to the term $p(s_i|\theta^s)$ in the Eq. 1.

It is well understood in the education research community that question difficulty and student proficiency are interdependent. Item Response Theory (IRT) [13] is used to estimate question difficulty and student proficiency jointly. The goals of the analyses of the question and student aspects of content creation are as follows:

- Organizing students into 3 proficiency categories (low, medium, and high) and the questions into 3 difficulty categories (easy, medium, and difficult), to direct the data collection process such that the eventual answer distribution is uniform/representative across the categories.
- Understanding overall characteristics of students and questions can be useful in refining their selection and creation processes, respectively.

Fig. 2 shows the question difficulty (QD) and student proficiency (SP) distributions. The more uniform the distributions, the better balanced are the selected questions and students. Some key observations with the large scale industry dataset are discussed below.

- **Medium proficiency heavy student selection:** The student proficiency distribution is non-uniform with significant mass concentrated in the middle. Thus, the students are not selected in a balanced approach. However, this distribution should be uniform to be able to collect the variety of answers.
- **Easy question:** There are about 8% questions which are very easy with β close to -2. Easy questions, in general, tend to have less variability in both the reference and student answers, making SRA training easier. Thus, less number of student answers for such questions should suffice for training.

Once a student set is selected, there is little that one can do to improve the student proficiency distribution as it is students' intrinsic property. Further, it

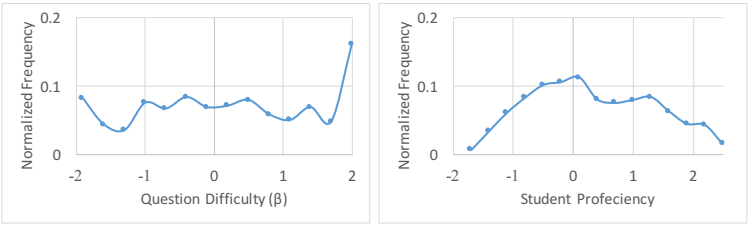


Fig. 2: (Left) Question Difficulty, where higher β corresponds to higher difficulty. (Right) Student Proficiency distribution.

Table 4: Distribution of student answers.

(a) Answers collected without considering question difficulty and student proficiency.

		Question Difficulty Bucket			
		Difficult	Medium	Easy	
Student Proficiency Bucket	**Low**	11.03%	9.27%	**6.83%**	27.13%
	Medium	15.88%	14.71%	13.69%	44.28%
	High	**8.30%**	9.12%	11.17%	28.59%
		35.20%	33.10%	31.70%	100.00%

(b) Answers collected in iteration-I.

		Question Difficulty Bucket			
		Difficult	Medium	Easy	
Student Proficiency Bucket	**Low**	**8.90%**	**7.20%**	**7.75%**	23.85%
	Medium	17.72%	11.51%	13.76%	42.99%
	High	10.93%	**9.51%**	12.71%	33.16%
		37.56%	28.22%	34.22%	100.00%

is difficult to exhaustively collect answers for each question from all students in a large scale data collection. Often, each question is answered by only a subset of students. Randomly allocating some students to answer a question may not lead to enough answer variants. This effect can amplify if all the students assigned a question are of similar proficiency. A potential mitigation strategy is to control how often a student with certain proficiency is asked a question of certain difficulty. For example, if there are more medium proficiency students, they should be asked a question less often as compared to low and high proficiency students. In this way, an imbalance of question difficulty and/or student proficiency can be prevented from reflecting in the answer distribution. Table 4a shows the distribution of the answers from the whole dataset, if collected without consideration of question difficulty and student proficiency. Note the imbalance in the distribution, specifically in terms of student proficiency. 44% answers are given by medium proficiency students. Particularly, SP:Low $\times$ QD:Easy is most affected. If the data is collected iteratively, this analysis can be useful in selecting the next set of student-question pairs. This assisted student answer collection form the basis of the iterative collection approach, as illustrated in Fig. 1b.

In the proposed data collection approach, the answers are collected in an iterative manner. In the first iteration $n^{a\prime}$ ($< n^a$) answers are collected from a random selection of students. At the end of the iteration, the answer distribution, as shown in Table 4, is computed. The less represented QD$\times$SP pairs are identified. This is followed by selecting students and questions belonging to desired SP and QD categories, respectively. The answers, thus collected, are cumulatively added to the overall dataset. The collection process is terminated after enough samples are collected and/or representativeness criteria of the dataset is satisfied.

To experimentally evaluate the effectiveness of the approach, we select $n^{a\prime} = 17$ randomly selected answers for each question in the first iteration. This results in selecting $\sim$50% answers, as there are $\sim$35 responses per question on average. Student proficiency and question difficulty are estimated from this subset. Ta-

Table 5: Results of utilizing the proposed iterative student selection approach. The data collected in proposed iterative approach is 28.1% smaller than the whole set. Results are reported in macro-average F1 and weighted F1 metrics.

	Training Size	Macro-F1	Weighted-F1
Base	13,169	57.39%	63.81%
After 1 Iteration	7,245	56.39%	63.25%
After 2 Iterations	9,470	**58.03%**	**64.20%**

ble 4b shows the observed distribution of the answers. For the next iteration, we select answers from only the following category combinations {SD:Low× QD:(Easy, Medium, Difficult), SD:High×QD:Medium}, as these categories are least represented in the collected answers. This way we add a total of 2,225 samples from the selected categories. At the end of each iteration, a classifier is trained and evaluated on a held-out test set. The classification results are reported in Table 5. We remove 681 `NON-ANSWER` responses from our entire dataset to create a 80-20 train-test split. *Base* represents the classifier performance after training on those 80% (13,169) samples. The number of samples after iteration-1 and iteration-2 are 7,245 and 9,470, respectively. Iteration-1 can be seen as a random subsample of the base set, whereas Iteration-2 benefits by inclusion of carefully chosen samples. After two iterations, the size of the dataset is 28.1% smaller than the base set; however, the macro-average F1 and weighted F1 are better than that of the base set.

6 Filter Student Answers

In a typical data collection setup, student answers are graded and then used as training samples. However, one can imagine that there may be outlier answers, which are not useful or, at times, misleading too. If we can automatically detect such answers *prior* to grading, corresponding human efforts can be saved. We propose following two approaches.

6.1 Automatically detect relatively less useful answers

The `NON-ANSWER` responses (e.g. 'I don't know', 'no idea') are not useful for training a classifier. Further, we find that student answers, even when incorrect, often tend to be from the same domain as the reference answer. Thus, we can filter out student answers that do not contain any domain keyword or terms in the corpus such as the textbook or relevant text material, yielding similar or better classifier model. When applying this filter on a combined set of 13,169 samples (*Base*) and 681 `NON-ANSWER` responses, we are able to filter out 785 responses, saving grading cost on 5.6% answers.

6.2 Very small and very long answers

A student answer which is too small is likely to miss out on most key aspects of the answer. Similarly, a very long answer may be the outcome of the student not knowing the exact answer. Both these types of answers are not likely to help significantly in training the classifier. Therefore, we propose to filter them out before grading to reduce the human efforts. For each question, we generate the distribution of the student response lengths and remove the samples not contained within $\mu \pm 2\sigma$, μ and σ being the mean and standard deviation of the student response lengths respectively. Using per question mean and standard

Table 6: Effect of discarding 5.6% training samples that are significantly different from reference answer; and, effect of length based pruning of 4.7% student answers; individually.

	Macro-F1	**Weighted-F1**
Base	57.39%	63.81%
− less useful	**60.02%**	**65.59%**
− $\|\mathbf{len} - \boldsymbol{\mu}\| \geq 2\boldsymbol{\sigma}$	**59.12%**	**65.04%**

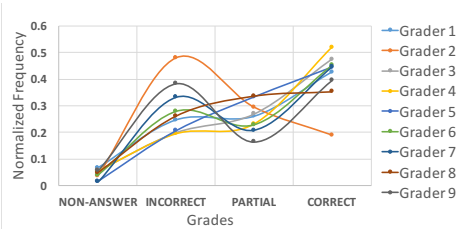


Fig. 3: Distribution of grades given by individual graders.

deviation statistics allows us to preserve question specific characteristics, as different questions can expect answers of different lengths. Using this approach, we are able to filter out 620 student responses from the *Base* training set (13,169 answers), leading to 4.7% reduction in grading cost.

Table 6 shows that the classification performance improves by individually rejecting the student answers based on their lengths as well as their differences from reference answers.

7 Filtering Graders

The graders grade student answers (a_i) in comparison to the reference answer (r_i) in context of the question (q_i). All the student answers pertaining to a specific question are assigned to exactly three graders; with each grader grading all those student answers. Thus, each student answer is given exactly three grades.

We aim to examine if there is any noticeable aspect (e.g. bias) in any grader's assessments. Fig. 3 shows the distribution of grades given by graders. The typical pattern is that the fraction of `CORRECT` grades is highest followed by `INCORRECT` and, then, by `PARTIAL` grade. Note that, a different pattern is exhibited by Grader–2 and Grader–8. Table 7 reports Pearson correlations pertaining to all grader pairs. Similar trends are observed for Cohen's Kappa and F1 metrics too, however they are not shown in this paper due to space constraints.

- Grader–2 has a significantly different grade distribution, with a peak at `INCORRECT`. Also, the `CORRECT` grade is at $\sim 20\%$ which is at least 15% less than all the other graders. Analyzing Table 7 reveals that Grader–2 has a very low agreement with other graders. It suggests that Grader–2 is a peculiarly strict grader. Perhaps, using the corresponding grades as labels can turn out to be misleading/noisy labels while training a classifier.
- Grader–8 appears to have a differently peculiar characteristic, with a relatively flat (`INCORRECT`: 26%, `PARTIAL`: 33%, and `CORRECT`: 35%) grading distribution. However, his/her agreement with other graders is not low. This suggests that Grader–8 may have been assigned easy-to-correctly-answer

Table 7: Inter-grader agreement in terms of Pearson Correlation. Each question is graded by three graders. For each grader pair, their mean agreement over the common set of assigned questions is reported.

	Grader 1	Grader 2	Grader 3	Grader 4	Grader 5	Grader 6	Grader 7	Grader 8	Grader 9
Grader 1				0.7514	0.6358	0.5965		0.6632	0.6811
Grader 2			0.6019	0.5353	0.5495		0.6432		
Grader 3		0.6019		0.6539		0.6533		0.6677	
Grader 4	0.7514	0.5353	0.6539		0.5630	0.6375		0.7827	0.6861
Grader 5	0.6358	0.5495		0.5630		0.6243	0.6257	0.6723	
Grader 6	0.5965		0.6533	0.6375	0.6243		0.7302	0.6362	
Grader 7		0.6432			0.6257	0.7302			
Grader 8	0.6632		0.6677	0.7827	0.6723	0.6362			0.6648
Grader 9	0.6811			0.6861				0.6648	

Table 8: Effect of discarding Grade–2's grades.

	Macro-F1	Weighted-F1
Base	57.39%	63.81%
After pruning Grader-2 (from only train set)	58.14%	63.60%
After pruning Grader-2 (from train and test sets)	**62.35%**	**66.13%**

questions. The chance of observing such unintentional bias may be rooted in question difficulty distribution and question assignment strategy.

Based on this analysis, we discard the grades given by Grader–2. Thus, the ground truths for the answers graded by Grader–2 are calculated as the majority vote of the other two graders. This led to the change of ground truths in $1,184$ samples in the train set and 232 samples in the test set. Table 8 shows the experimental results. We test with two configurations - pruning Grader–2 grades from only train set and from both train and test set. Results show that while both lead to improvements in the test set Macro-average F1s, the latter performs significantly better.

8 Conclusion

This paper focuses on reducing the human-efforts while collecting dataset for training student response analyzer. We begin by outlining the stages of data collection, its generative modeling, and the cost associated. Unlike many tasks where the cost of dataset creation is mostly related to labeling, the dataset creation for student response analyzer includes the significant cost for question creation and student answer collection along with labeling. We propose an iterative approach to significantly reduce the human-efforts, while improving the classification performance. That leads us to first proposing a deep learning based approach to predict question difficulty prior to answer collection. To ensure a good spread of collected data, we also propose a technique to select students to answer certain questions; and, thus enabling a focused data collection. Further, we show that a significant portion of student answers can be filtered out saving the cost of grading them while improving the classification performance by up to 2% in macro-average F1. By pruning out poor quality graders, our classification result improves up to 5% in macro-average F1. Finally, our focused answer collection approach saves up to 28% in answer collection cost and answer filtering saves $\sim$10% of grading cost.

References

1. Arora, S., Nyberg, E., Rosé, C.P.: Estimating annotation cost for active learning in a multi-annotator environment. In: Proceedings of the NAACL-HLT Workshop on Active Learning for Natural Language Processing. pp. 18–26 (2009)
2. Baldridge, J., Osborne, M.: Active learning and the total cost of annotation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (2004)
3. Basu, S., Jacobs, C., Vanderwende, L.: Powergrading: a clustering approach to amplify human effort for short answer grading. Transactions of the Association for Computational Linguistics 1, 391–402 (2013)
4. Bernstein, M.S., Little, G., Miller, R.C., Hartmann, B., Ackerman, M.S., Karger, D.R., Crowell, D., Panovich, K.: Soylent: a word processor with a crowd inside. Communications of the ACM 58(8), 85–94 (2015)
5. Birnbaum, A.: Some latent train models and their use in inferring an examinee's ability. Statistical theories of mental test scores pp. 395–479 (1968)
6. Brooks, M., Basu, S., Jacobs, C., Vanderwende, L.: Divide and correct: using clusters to grade short answers at scale. In: Proceedings of the ACM conference on Learning@ scale conference. pp. 89–98 (2014)
7. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 670–680 (2017)
8. Dzikovska, M.O., Nielsen, R.D., Brew, C.: Towards effective tutorial feedback for explanation questions: A dataset and baselines. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 200–210 (2012)
9. Guan, D., Yuan, W., Ma, T., Khattak, A.M., Chow, F.: Cost-sensitive elimination of mislabeled training data. Information Sciences 402, 170–181 (2017)
10. Gweon, G., Rosé, C.P., Wittwer, J., Nueckles, M.: Supporting efficient and reliable content analysis using automatic text processing technology. In: Proceedings of IFIP Conference on Human-Computer Interaction. pp. 1112–1115 (2005)
11. Horbach, A., Palmer, A., Wolska, M.: Finding a tradeoff between accuracy and rater's workload in grading clustered short answers. In: International Conference on Language Resources and Evaluation. pp. 588–595 (2014)
12. Hsueh, P.Y., Melville, P., Sindhwani, V.: Data quality from crowdsourcing: a study of annotation selection criteria. In: Proceedings of the NAACL-HLT Workshop on Active Learning for Natural Language Processing. pp. 27–35 (2009)
13. Johnson, M.S., et al.: Marginal maximum likelihood estimation of item response models in r. Journal of Statistical Software 20(10), 1–24 (2007)
14. Jurgens, D.: Embracing ambiguity: A comparison of annotation methodologies for crowdsourcing word sense labels. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 556–562 (2013)
15. Kittur, A., Smus, B., Khamkar, S., Kraut, R.E.: Crowdforge: Crowdsourcing complex work. In: Proceedings of the ACM symposium on User Interface Software and Technology. pp. 43–52 (2011)
16. Kulkarni, A., Can, M., Hartmann, B.: Collaboratively crowdsourcing workflows with turkomatic. In: Proceedings of the ACM Conference on Computer Supported Cooperative Work. pp. 1003–1012 (2012)
17. Nicholson, B., Sheng, V.S., Zhang, J.: Label noise correction and application in crowdsourcing. Expert Systems with Applications 66, 149–162 (2016)

18. Nicholson, B., Zhang, J., Sheng, V.S., Wang, Z.: Label noise correction methods. In: Proceedings of IEEE International Conference on Data Science and Advanced Analytics. pp. 1–9 (2015)
19. Rosé, C.P., Moore, J.D., VanLehn, K., Allbritton, D.: A comparative evaluation of socratic versus didactic tutoring. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 23 (2001)
20. Sánchez, J.S., Barandela, R., Marqués, A.I., Alejo, R., Badenas, J.: Analysis of new techniques to obtain quality training sets. Pattern Recognition Letters 24(7), 1015–1022 (2003)
21. Settles, B., Craven, M., Friedland, L.: Active learning with real annotation costs. In: Proceedings of the NIPS Workshop on Cost-Sensitive Learning. pp. 1–10 (2008)
22. Sheng, V.S., Provost, F., Ipeirotis, P.G.: Get another label? improving data quality and data mining using multiple, noisy labelers. In: Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining. pp. 614–622 (2008)
23. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 254–263 (2008)
24. Valizadegan, H., Tan, P.N.: Kernel based detection of mislabeled training examples. In: Proceedings of the SIAM International Conference on Data Mining. pp. 309–319 (2007)
25. Zesch, T., Heilman, M., Cahill, A.: Reducing annotation efforts in supervised short answer scoring. In: Proceedings of the NAACL-HLT Workshop on Building Educational Applications. pp. 124–132 (2015)