Can Students Produce Effective Training Data to Improve Formative Feedback?

Yulin Zhang
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
yzhan114@ncsu.edu

Edward F. Gehringer
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
efg@ncsu.edu

***Abstract*—This full research paper shows how machine learning can improve peer assessment by giving students advice on how to write better quality reviews. We trained a model that gives automated feedback by using labeled data produced by students over a period of several semesters. To improve the accuracy of the model, we are working to incorporate active learning (in the machine-learning sense) to direct students to produce training data for situations where the model has the most difficulty making predictions. With the active-learning approach, we expect students to have to do less labeling, so that they can be more attentive and produce more accurate labels. Our results revealed that we are able to cut the amount of labeling effort by half, without loss of reliable training data.**

***Index Terms*—Machine learning, active learning, peer assessment, peer review, automated feedback**

I. Introduction

Peer assessment can be a valuable learning experience. The reviews that students write, and those that they receive can help clarify for them the characteristics of a good piece of work. To write good peer reviews, students need to learn what makes a review effective. Our approach is to utilize our machine-learning system in giving students instant feedback on the quality of their reviews. This helps dissuade students from submitting poor reviews, as they can take into account the feedback before submitting the review.

After submitting their work, students are asked to review the work of a few of their peers. Reviews are based on a rubric, which contains a set of items for each of which the reviewer is asked either to assign a score, give a textual feedback comment, or both, as shown in Figure 1. After filling out the rubric, the reviewer is presented with instant feedback, and can factor it into the review. Reviewees, whose work is being reviewed, then have the opportunity to improve and resubmit their work based on the reviews they receive. Depending on how the assignment is set up, the reviewer may be asked to review the work a second time, to judge whether the recommendations from the first review have been acted upon. Through this iterative process, students are induced to think meta-cognitively about the work they are doing, and they thereby gain a deeper understanding of the subject matter.

The machine-learning model that provides reviewers with the instant feedback is trained using labeled data from actual reviews submitted by students. After the assignment is over, the reviewees are invited to label each comment in the reviews they have received, as containing (or not containing) certain characteristics. For example, the reviewees might be asked to decide whether a review comment contains a suggestion, whether it detects a problem, whether it has positive tone, etc.

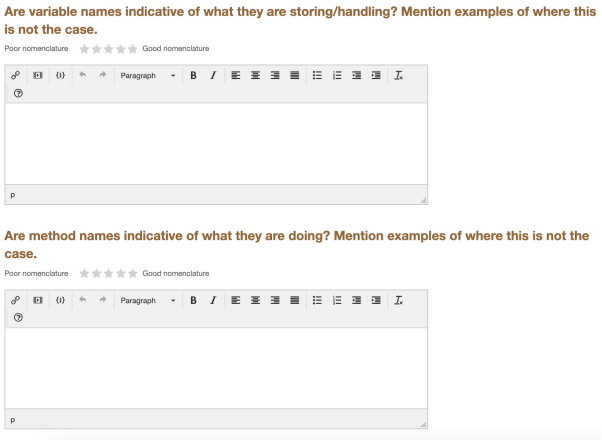


Fig. 1: Student reviewers are presented with rubric items which they are asked to respond to with numeric scores and textual comments.

The labels that the students produce can be used as training data for a machine-learning model that will automatically assess student-submitted reviews. Reviews might be judged as "better," for example, if they detect more problems and make more suggestions. The resulting judgments can then be conveyed to the student reviewer who is about to submit the review, to encourage him/her to improve the review, and/or conveyed to the teaching staff, who can use it to grade students on their reviewing.

When this labeling approach was first used, students were asked to label hundreds of review comments. Researchers were concerned that this amount of work could lead inattentive participants to submit unreliable labels, thereby undermining the effectiveness of the machine-learning model. Active learning (in the machine-learning sense) aims to address this concern by having the machine-learning model pre-determine the model's confidence level that the comment contains the characteristic (suggestion, problem detection, positive tone, etc.) and only

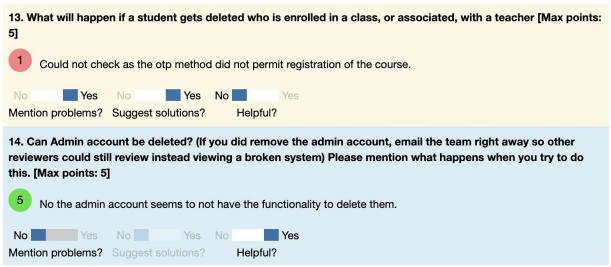


Fig. 2: Screenshot of a page labeled by a student. Students are required to provide input to label prompts with a white slider track. The student can answer "Yes" or 'No' to each label prompt by toggling the slider. With active learning incorporated, students see two additional types of label prompts: the grayed-out label prompt with reduced opacity and the label prompt that has received conflicting answers from the machine and student.

asks the student to manually label a comment when the confidence level is below an assigned certainty threshold. For example, the student might be asked to label a comment if the machine has less than 65% confidence that it does (or does not) contain a suggestion.

The first half of this paper centers on two points of investigation:

- For a given level of confidence, what fraction of the original number of comments does a student still need to label?
- For a desired level of labeling accuracy, how high does the label certainty threshold need to be? For example, if we desire 90% of the comments that are labeled by the machine to be labeled correctly, how high do we need to make the label certainty threshold?

The second half of the paper explores the effect that reducing the labeling workload has on the quality of student-assigned labels. This has implications for whether the active-learning approach can improve the quality of labels for the machine-learning model.

II. Related Work

The earliest work in the area of automated feedback for textual peer-review feedback appears to be Nguyen et al. [1]. It uses a natural language processing strategy to detect the presence of "solutions" in peer reviews. Ramachandran et al. [2] developed an "automated meta-review" approach for rating submitted reviews just before they are submitted. Both of these approaches provide instant feedback to the reviewer, who can then improve the review before submitting it. Both of these works use a rule-based approach, in contrast to the machine-learning approach investigated in this paper. Instead of focusing on the textual feedback in a review, Rico-Juan et al. [3] focus on summative grades assigned by students in a peer-review system, and attempt to detect bias in the evaluation. Our previous work in this area involves machine-learning strategies for recognizing suggestions [4] and detecting problems [5] in the reviewed work. Recently, Darvishi et al. [6] presented a "learnersourcing" approach to evaluate learning materials based on student feedback, that takes into account both scores and textual comments.

III. Method

This study examines data collected between the Fall 2018 semester and the Spring 2021 semester from a masters-level course in software engineering. After a peer-reviewed assignment was finished, students were offered extra credit for assigning labels to comments that *their* peer reviewers had assigned to *their* work, as containing or not containing certain characteristics that a good review should have. Then, depending on how teaching staff perceived the quality of their labels, students who had labeled comments received full or partial credit on a 10-point scale. We investigated the labels produced by students as well as the grades assigned to these labels from both the current semester and previous semesters.

- In Spring 2021, for full credit, we asked students to assign the 200 labels that the machine learning models were most uncertain of.
- In Fall 2020, for full credit, we asked students to assign any 200 labels to the comments that were available to be labeled.
- In earlier semesters (Fall 2018 to Spring 2020), for full credit, we asked students to label all the comments received on their reviews.

As stated in the Introduction, the incorporation of active learning involves pre-loading machine-learning predictions on what labels to assign to comments, and then only asking the student to manually label a comment when the confidence level is below the chosen label certainty threshold. The confidence level is necessarily between 0.5 and 1, because if we are less than 50% certain that a comment contains a suggestion, for example, then we are more than 50% certain that it does not contain a suggestion. If the label's confidence level is above or equal to the threshold, it is considered reliable and therefore is grayed out in the labeling view shown to the student (see Figure 2). This indicates that a student is not required to assign a label, but may do so if they think that the label predicted by the model is incorrect.

To answer the question, For a given level of confidence, what fraction of the original number of comments does a student still need to label? we set the certainty threshold to a level of 0.5, and then repeatedly incremented it by 0.05 until it reached 1.0. At each level, we recorded how many machine predictions had a confidence level that was at or above the certainty threshold. (Naturally, the number of predictions with the required confidence level decreases as the certainty threshold increases.) We also recorded at each level the percentage of labels where the student and the machine disagreed. By "disagree," we mean that the machine assigned the label differently than the student, for example, one labeled "Yes, the characteristic is present," and the other labeled "No, the characteristic is not present."

We selected the percentage of labels in disagreement as our metric for two reasons. First, assuming the model was highly capable of making the right predictions, then the percentage of disagreeing labels tells how many mistakes students made under the imposed conditions. Second, assuming the majority of students' labels are dependable, the percentage of disagreeing labels serves as an estimate of the machine's accuracy at a given level of certainty. This answers our second question, Given an error in automatically labeling a particular comment, how high would the level of confidence have to be so that it was not incorrectly labeled? This metric suggests a reasonable threshold level to use in subsequent experiments to investigate the effects further. And this metric works well, as the machine is less subjective in evaluating the comments than humans are, and therefore more consistent.

This study measures the quality of students' labeling in two ways. One is to directly refer to the grades teaching staff gave to students for each completed labeling assignment. From the set of assigned grades, we eliminated grades from students who did not participate in the labeling assignment, and divided each of the remaining grades by the maximum grade the student could obtain for the number of labels they assigned (for example, if a student who was required to assign 200 labels for full credit ended up assigning 60 labels, his/her maximum grade would be $60 \div 200 \times 10 = 3$). Then we calculated the average of (maximum grade student could receive / grade student actually received) over all students. The resulting value will be referred to as the *human-assigned grade* for labeling that assignment.

Another way to assess labeling quality is to take the number of labels in disagreement and divide it by the total number of labels assigned by students. This gives us the percentage of labels that were in disagreement between human and machine. We subtracted this value (percentage of labels in disagreement) from 1 and multiplied it by 10 to get what we will call the *machine-assigned grade*. We studied the changes over time in human-assigned grades and machine-assigned grades to gain a comprehensive understanding about which semester and which assignment students performed the best.

IV. Results

There were two convolutional neural network (CNN) models used in this study, one that recognizes when a review detects a problem in the work, and one that recognizes when a review makes a suggestion. When presented with a piece of review text, the model returns the probability that the review text detects a problem or contains a suggestion. Ultimately, we want to count the number of comments in each review that detect problems or contain suggestions. Then we can give feedback to the reviewer on how their review measures up against a standard, or against reviews submitted by their classmates. But how certain should the model have to be that there is a problem or a suggestion before it counts it as one? If we require greater certainty, then the model will count fewer problem detections or suggestions. If we relax our certainty requirement, then the model will infer that more comments detect problems or contain suggestions. Let us call the level of certainty that we require in order to count a problem detection or a suggestion the *label certainty threshold*.

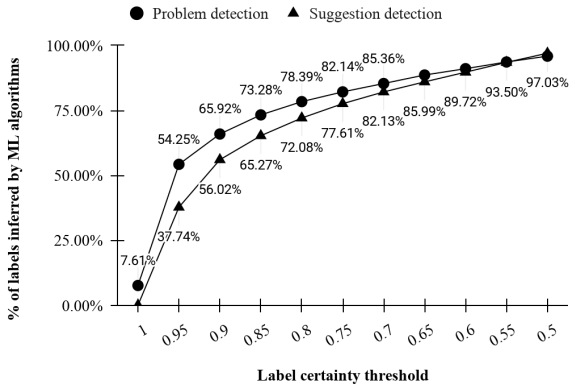


Fig. 3: The percentage of labels inferred by the ML models increases as the label certainty threshold declines. The most drastic increase occurs between threshold levels 1 and 0.9. Below that the slope is less drastic.

Figure 3 shows how the percentage of inferred labels changes as the label certainty threshold declines. One can see that as the threshold was diminishing, the percentage of inferred labels kept increasing, and it increased most dramatically between threshold levels of 1 and 0.9. At the 0.9 level, both the problem-detection and suggestion-detection models were able to infer more than half of the total number of labels, with the problem detection model hitting almost 70% in coverage.

Recall that inferred labels are reliable labels for which we do not require students to provide their input (unless they decide that they are incorrectly assigned). We can say that we reduce students' workload by giving them less labeling work to do. We can interpret the *y*-axis to be % effort reduction for our students as we change the label certainty threshold. For example, we will be able to reduce the workload by half by setting the label certainty threshold to 0.9.

Figure 4 shows the percentage of inferred labels disagreed with by students, as the label certainty threshold changes. One can see that both trendlines are increasing, with the steepest increase occurring between threshold levels 1 and 0.9, which is the same range identified in Figure 3. Below 0.9 the number of inferred labels not disputed by students increases little, probably because about a half of labels were already inferred at the threshold level of 0.9, so there is not much room for increase. We observed that students and the machine had more conflicting opinions on how the suggestion labels should be assigned than on how the problem labels should be assigned. Based on the label assignments we have seen, it appears to us that students and the machine both contribute to the disagreements in some way, either due to the carelessness of students during labeling or the inaccuracy of machine predictions themselves. It would be worthwhile to

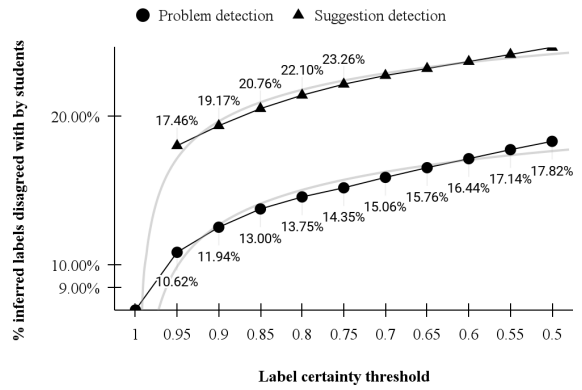


Fig. 4: The percentage of inferred labels where students disagree with the machine increases as the label certainty threshold declines. Inferred suggestion labels seemed to be disagreed more by students than inferred problem labels. This figure resembles the trend found in Figure 2 except that the two trends did not converge.

further analyze which of these two causes a greater impact.

For the following analyses, we set the label certainty threshold to be 0.9, assuming that labels whose certainty scored at or above this level were mostly inferred correctly.

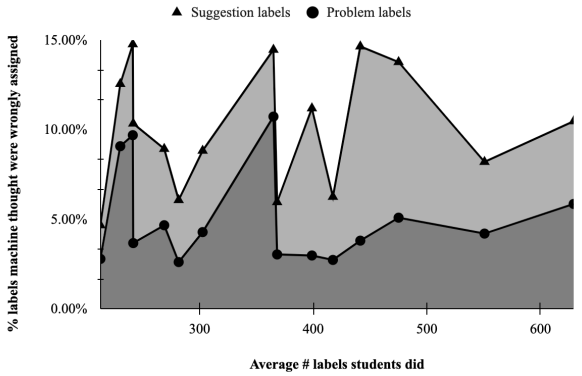


Fig. 5: The percentage of labels that the machine determined to be wrongly assigned by students does not rise noticeably as the average number of labels assigned by students increases.

Figure 5 shows how the "student error rate" of label assignment changes as the average number of labels students assigned increases. The student error rate is the fraction of errors determined by the machine to be wrongly or inattentively assigned. We acknowledge that the machine can sometimes make wrongful judgments, but we expect its "machine error rate" to be constant, so the changes in the number of wrongly assigned labels should be attributed mainly to the student side. Our hypothesis was that, since students have a limited amount of energy to devote to repetitive tasks, if we add to their workload, their performance drops. However, from Figure 5 we do not notice an increase in the number of wrongly assigned labels as the number of assigned labels increases, though the fluctuation is quite large. The figure is not convincing enough to discern the relationship between the two, and hence further analysis is needed.

We had students label review comments in semesters between Fall 2018 and Spring 2021. In each case, they labeled reviews for three assignments, Program 2 (a Rails application where all teams wrote the same code), OSS Project (an open-source software project where different teams worked on different parts of the code), and Design Doc, on the design of students' (different) final projects.

TABLE I: Labeling statistics for three assignments in each of six semesters.

Semester	Assignment	# Students that participated in labeling	Average # labels that students produced	Grade given to students based on quality of their labels
F18	Prog 2	66	213	N/A
	OSS	74	368	
	DesDoc	72	399	
S19	Prog 2	54	282	9.71
	OSS	55	417	9.95
	DesDoc	52	441	9.96
F19	Prog 2	95	303	8.49
	OSS	94	551	8.58
	DesDoc	90	475	9.24
S20	Prog 2	33	365	9.36
	OSS	36	241	9.20
	DesDoc	26	230	8.66
F20	Prog 2	39	629	6.89
	OSS	49	269	8.55
	DesDoc	44	242	8.91
S21	Prog 2	13	498	10.0
	OSS	13	192	9.84
	DesDoc	10	244	8.90

The number of students in 2020-2021 was greatly reduced due to the pandemic.

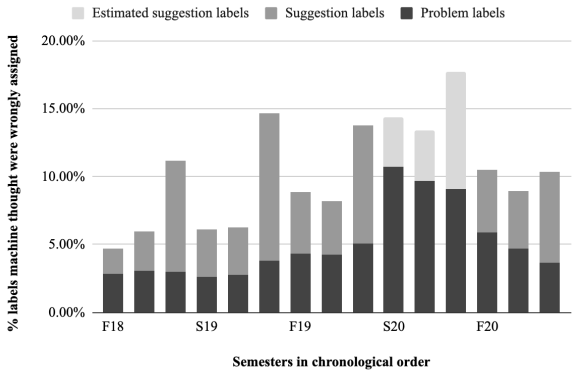


Fig. 6: The percentage of labels that the machine determined to be wrongly assigned by students in S20 is the greatest among all semesters, while among all assignments of a semester, the percentage disagreement of the third assignment (Design Doc) is the greatest.

Surprisingly, as shown in Figure 6, we found that instead of

the number of labels that students assigned, the semester they were in and the project for which they were labeling comments were associated with greater differences in accuracy. In Spring 2020, we did not ask students to label for the presence of suggestions, so we used the average of all other semesters as an estimate. We see that in the Spring 2020 semester, the machine determined the percentage of erroneous labels to be highest, while for other semesters the percentage of erroneous labels was generally more equal. However, as we inspected more closely, we found explanations that might be related not to the student's performance, but to other external factors like the way rubric prompts were crafted.

For example, consider the two questions below. Without knowing how the prompt is phrased, the machine would base its judgment solely on the comments. From the machine's perspective, "Admin account cannot be deleted" and "I am not able to delete the user to test this functionality" are likely the same and should be judged the same way. Students when doing their labeling, and teaching staff when doing their grading, were exposed to more contextual information and therefore are more likely to make correct judgments. In the first example, the assignment specifications say that the administrator account should not be able to be deleted. Thus, when the student reviewer says that it can't be deleted, the review is not describing anything wrong with the program. The machine, however, does not "understand" this context, and thus guesses that the review comment is indicative of a problem. More such examples can be found in the appendix.

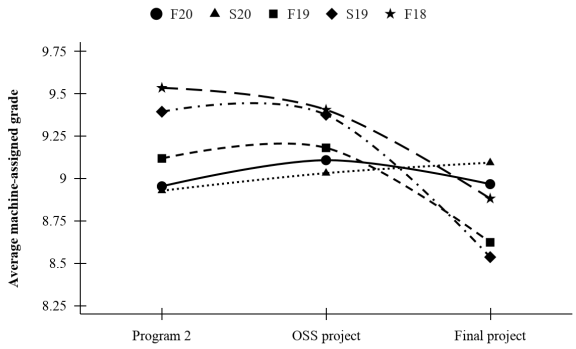
Question 1

Q: Can Admin account be deleted? Please mention what happens when you try to do this.
A: Admin account cannot be deleted
Machine labeled: Yes, mentioned problem
Student labeled: No, did not mention problem

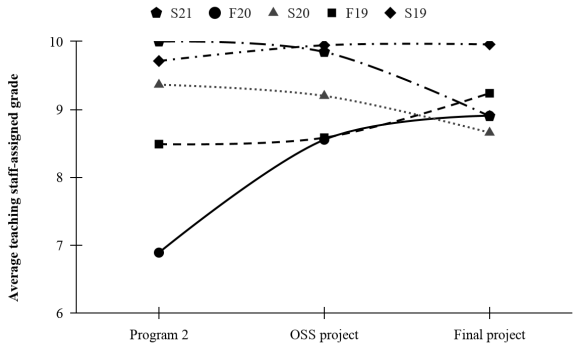
Question 2

Q: What happens when the admin tries to delete a user who has multiple pending special item requests? Do the requests get deleted?
A: I am not able to delete the user to test this functionality.
Machine labeled: Yes, mentioned problem
Student labeled: Yes, mentioned problem

Another pattern we see from Figure 6 is the machine's inability to produce correct labels for the last assignment in the course, the Design Document. The percentage of errors is nearly double, if not triple, compared to previous assignments. This could be because each design doc related to an independent project, unlike Program 2, where all teams coded the same project. We located the places where the disagreements occurred, and we discovered that for the Design Document, our suggestion model seemed to be over-optimistically inclined to classify comments as containing suggestions, as most disagreements occurred when the machine assigned "Yes" and students assigned "No", whereas for the previous assignments, the suggestion detection model produced fewer "Yes" labels and therefore disagreements were less.



(a) Average machine-assigned grades to students' labels.



(b) Average human-assigned grades to students' labels.

Fig. 7: The machine determined that the produced labels became less accurate as the semester progressed, while the teaching staff did not observe this falloff in accuracy.

Figure 7 shows the average grade the machine (a) and the teaching staff (b) gave to each labeling assignment. The human-assigned grades were directly drawn from de-identified spreadsheets furnished by the instructor, while the machine-assigned grades were calculated as:

$$10 \times (1 - w)$$

where 10 is the maximum number of points students can earn for a labeling assignment, and w is the fraction of labels for that assignment that were incorrectly assigned, according to our machine-learning models.

To test the validity of using labels produced by the machine models to compare against students' labels, we observed whether the patterns in Figure 7(a) and Figure 7(b) look alike. The result does not confirm our expectations. We see that while the machine found that the quality of student-produced labels worsened as the semester progressed, the teaching staff thought it was improving, a direct contradiction. We suspect that human-assigned grades are more likely to be valid, because humans actually understand the text, and can take into account additional information, such as the prompt. Moreover, as students were learning from the feedback they

get from teaching staff on their labeling of earlier assignments, they tuned their behaviors to match with the *teaching staff*'s expectations rather than tuning them to match the machine algorithm. This may explain the improvement in human-assigned grades.

V. Discussion

A. Does reducing students' workload lead to better quality labels?

Does reducing students' workload lead to better quality labels? We believe the answer to be yes. It is not because students know ahead of time that the amount of labeling will be daunting, so they decide to do a poor job, but rather that we trust students to treat the task seriously no matter how daunting it sounds. However, after labeling many comments, they find it difficult to concentrate, and may consider less evidence in assigning labels. The goal of our study is to determine the threshold where students get exhausted and start to make mistakes.

If we were looking for the semester when the students felt most overwhelmed, it might be Spring 2020. As shown in Table II, in previous semesters, the students had been asked to label no more than ten rubric items per review. However, in Spring 2020, we gave the students 21 rubric items per review to label. We found that Spring 2020 students did the least amount of labeling work per assignment, and both the machine and teaching staff graded this semester mediocrely. This was the semester with the greatest workload, and ironically, the one where students did the least work. Starting with Fall 2020, we imposed a ceiling of 200 on the number of comments students would be asked to label, and labeling quality went back up.

TABLE II: Number of rubric items per review in each semester

Semester	Number of rubric items per review
Fall 2018	10
Spring 2019	10
Fall 2019	10
Spring 2020	21
Fall 2020	50
Spring 2021	41

Note that doing less labeling work was not always associated with high-quality labeling. This suggests that students may have vague ideas about how labels are to be assigned. They may misunderstand what the label prompt asks, or they have their own understanding of these prompts that is different from how the teaching staff would interpret them.

For example, in the Fall 2020 semester when we asked for helpfulness labels, students responded differently to what we would expect. While teaching staff thought brief comments like "Yes" are not helpful for their lack of constructive content, some students perceived them to be helpful because such comments reassured them that their code works.

Likewise, considering the following comment:

"There is no destroy button for me to remove an entry from the Teacher's table."

For some students, the text can be seen as containing suggestions if by reading the text they know immediately how to fix the problem. One can argue that this comment is suggesting to add a destroy button to the Teacher's table, although not explicitly, while others can argue only the explicit suggestion counts. In other words, labeling peer-review comments can be fairly subjective in that it largely depends on students' own interpretation of the comments.

B. Active learning, will it work?

While conducting the research, we observed a point that we feel might be worth mentioning here. The role that active learning plays in the labeling process can be more than just reducing the number of label queries. Active learning will not only make students do less labeling but also keep them attentive to the labeling process. Students need to constantly scroll to the next unlabeled comment, as there may be several labels in between that have already been inferred by the machine. And as students are jumping among labels, they observe how much of the work has already been judged by the machine, which may encourage them to fill in the knowledge gaps of the machine.

In the Spring 2021 semester, we put active-learning code to use and observed quite promising results: all students produced labels that the teaching staff team determined to be accurately assigned. However, since only 13 students assigned labels this semester (the class was small due to the unavailability of student visas during the pandemic), the number of participants is not sufficient to build a good case for the efficacy of active learning in the labeling process. Further study should continue to examine the benefits of active learning.

VI. Conclusion

Peer assessment promotes an effective and collaborative learning environment. It helps students to explore opportunities for improvement, and by showing their work to others they can more easily expose imperfections in their work, receive advice and encouragement, and ultimately lead to better learning outcomes. By incorporating active learning into this process, the two parties, machine and human, can positively influence each other. Students can learn to write better-quality comments to their peers under the instant feedback given by the machine to ensure all the characteristics of good reviews are presented in their comments, while the model can be better trained using more accurate labeling inputs from students.

This study explores the estimated accuracy of the two selected machine-learning models, the problem-detection model and the suggestion model, as well as the impact of workload on label quality. Our results indicate that at a label certainty threshold level of 0.9, both models were able to infer more than half of the labels that students originally had to generate, with an acceptable disagreement rate (between students' labeling and labels inferred by the model) of 11.94% among problem labels and 19.17% among suggestion labels. Setting the threshold level to 0.9 minimizes the workload of students without causing them to doubt the results of the

model. Unfortunately, our study fails to identify the direct association between workload and label quality, since the percentage of labels in disagreement did not increase as the workload became heavier. We speculate that with more experimental data, the effect of reduced workload will become more apparent.

Regardless, our study does demonstrate that, in the realm of peer assessment, students can be relied upon to produce a large volume of training data, with a quality level that would be difficult to improve upon, even by paying participants. This suggests that in other areas of learning technologies, such as student-generated exercises or intelligent tutoring, labeling by students can help by providing the data to develop machine-learning models.

REFERENCES

[1] Nguyen, H., Xiong, W., & Litman, D. (2016, June). Instant feedback for increasing the presence of solutions in peer reviews. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations* (pp. 6-10).

[2] Ramachandran, L., Gehringer, E. F., and Yadav, R. "Automated assessment of the quality of peer reviews using natural language processing techniques," *International Journal of Artificial Intelligence in Education*, January 2017, pp. 1-48.

[3] Rico-Juan, J. R., Gallego, A. J., Valero-Mas, J. J., & Calvo-Zaragoza, J. (2018). Statistical semi-supervised system for grading multiple peer-reviewed open-ended works. *Computers & Education*, 126, 264-282.

[4] Zingle, G., Radhakrishnan, B., Xiao, Y., Gehringer, E. F., Xiao, Z., Pramudianto, F., Khurana, G. and Arnav, A. "Detecting suggestions in peer assessments," EDM 2019: 12th International Conference on Educational Data Mining, Montreal, July 2019, pp. 474-479..

[5] Yunkai Xiao, Gabriel Zingle, Qinjin Jia, Harsh Shah, Yi Zhang, Tianyi Li, Mohsin Karovaliya, Weixiang Zhao, Yang Song, Jie Ji, Ashwin Balasubramaniam, Harshit Patel, Priyankha Bhalasubbramanian, Vikram Patel and Edward Gehringer, "Detecting Problem Statements in Peer Assessments," EDM 2020: 13th International Conference on Educational Data Mining, July 2020, pp. 704-709.

[6] Darvishi, A., Khosravi, H., & Sadiq, S. (2020, September). Utilising Learnersourcing to Inform Design Loop Adaptivity. In *European Conference on Technology Enhanced Learning* (pp. 332-346). Springer, Cham.

APPENDIX

Here are more examples of labels in disagreement. The answers that are in bold are what our authors think is correct.

Question 1

Q: Are the teachers able to add or remove themselves from a class.

A: Unable to remove an entry from the Teache[r']s table, probably just forgot to add the destroy button there?

Machine labeled: No, did not suggest solutions

Student labeled: **Yes, suggested solutions**

Question 2

Q: Are there comments for each method? Are they descriptive enough to tell what the method does, and how to does it?

A: A small number of methods do not have method comments, and most of these are simple methods whose function can be easily deciphered.

Machine labeled: Yes, suggested solutions

Student labeled: **No, did not suggest solutions**

Question 3

Q: Do you think that this code is ready to be deployed onto the production server (for the corresponding OSS project)? If not, what are your biggest concerns? ...

A: Well tested, following current structural patterns, could be considered for deployment

Machine labeled: Yes, mentioned problem and suggested solutions

Student labeled: **No, did not mention problem nor suggest solutions**

Question 4

Q: Is the teacher able to log in with his/her email id and password?

A: Cannot login with the teacher credentials provided, just redirects to login page.

Machine labeled: **No, did not suggest solutions**

Student labeled: Yes, suggested solutions

Question 5

Q: Overall, how good is the write-up? If you found problems with the write-up (lack of explanations of the functionality, lack of explanation of how to check ...)

A: Easy to read and with pictures inserted as examples. But some of the pictures could be smaller and the introduction part is kind of redundant in my mind.

Machine labeled: **Yes, suggested solutions**

Student labeled: No, did not suggest solutions

Question 6

Q: Did the team add test cases? Did the coverage increase? How well do the newly added tests cover the range of this project?

A: Yes, test cases were added and the overall coverage increased! the tests did a good job of insuring that despite refactoring everything still functioned as before.

Machine labeled: **No, did not suggest solutions**

Student labeled: Yes, suggested solutions