Attitudes Surrounding an Imperfect AI Autograder

Silas Hsu*
Tiffany Wenting Li*
silash2@illinois.edu
wenting7@illinois.edu
Department of Computer Science,
University of Illinois at
Urbana-Champaign, United States

Zhilin Zhang
zhilinz2@illinois.edu
Department of Computer Science,
University of Illinois at
Urbana-Champaign, United States

Max Fowler
mfowler5@illinois.edu
Department of Computer Science,
University of Illinois at
Urbana-Champaign, United States

Craig Zilles
zilles@illinois.edu
Department of Computer Science,
University of Illinois at
Urbana-Champaign, United States

Karrie Karahalios
kkarahal@illinois.edu
Department of Computer Science,
University of Illinois at
Urbana-Champaign, United States

ABSTRACT

Deployment of AI assessment tools in education is widespread, but work on students' interactions and attitudes towards imperfect autograders is comparatively lacking. This paper presents students' perceptions surrounding a ~90% accurate automated short-answer grader that determined homework and exam credit in a college-level computer science course. Using surveys and interviews, we investigated students' knowledge about the autograder and their attitudes.

We observed that misalignment between folk theories about how the autograder worked and how it actually worked could lead to suboptimal answer construction strategies. Students overestimated the autograder's probability of marking correct answers as wrong, and estimates of this probability were associated with dissatisfaction and perceptions of unfairness. Many participants expressed a need for additional instruction on how to cater to the autograder. From these findings, we propose guidelines for incorporating imperfect short answer autograders into classroom in a manner that is considerate of students' needs.

CCS CONCEPTS

• **Human-centered computing → Empirical studies in HCI**; *Natural language interfaces*; • **Applied computing → Interactive learning environments**; • **Computing methodologies →** Natural language processing.

KEYWORDS

human-AI interaction, imperfect AI, perception and acceptance of AI, folk theories, autograder, computer science education, ASAG, EiPE, code reading

ACM Reference Format:
Silas Hsu, Tiffany Wenting Li, Zhilin Zhang, Max Fowler, Craig Zilles, and Karrie Karahalios. 2021. Attitudes Surrounding an Imperfect AI Autograder. In *CHI Conference on Human Factors in Computing Systems (CHI'21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3411764.3445424

1 INTRODUCTION

Computers have been used to automatically assess and grade students' programming assignments since the 1960s [23]. Today, computers assess students in various situations, from grading multiple-choice questions, to evaluating formulas in mathematics homework [29], to providing feedback on essays [27]. As class sizes increase, automatic grading increasingly contributes to savings in human labor and timely feedback [3, 27, 29, 55].

In the past two decades, AI-powered autograders that grade natural language responses to short-answer questions and essay prompts have gained prominence [10, 48, 53]. Natural language processing (NLP) autograders have been deployed both in K-12 settings [34, 54] and standardized high-stakes settings including the Graduate Record Examinations (GRE) in the US [2]. Modern systems that automatically evaluate student writing have a high agreement with human raters [2, 9, 47], yet this has not resolved all controversy related to their deployment [8, 12, 14, 50].

Outside of the education domain, research has shown how algorithms' imperfections and differences from human judgement may result in trust and acceptance issues [17, 33, 36], but we know less about how students interact with and perceive imperfect AI autograders. Forming guidelines on how to manage students' and teachers' perceived accuracy, fairness, educational value, and other attitudes towards NLP autograders will be key to further adoption of these systems. In other words, as Williamson et al. states [55], researchers should no longer focus on "can it be done?" but "how should it be done?" We tackle this question in the context of automated short answer grading (ASAG). To the best of our knowledge,

[*] Both authors contributed equally to this research.

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445424

we are the first to investigate students' perceptions of ASAG system functionality (e.g., accuracy, fairness, educational value) and how students cater their answers to autograders, which could affect the assessments' validity. Moreover, the automated classroom ecosystem lacks guidelines or good practices for using AI to assess free-form responses.

This paper presents a mixed-methods study (surveys + interviews) of students' interactions with an ASAG system used on homework and exams in a full-semester introductory college computer science course. The ASAG system specifically graded responses to *code reading* questions. These are questions that require students to explain a piece of code in plain English. Points awarded by the algorithm mattered; they contributed to students' final grades. At the same time, the algorithm's false positive and false negative rates were 15% and 10%, respectively. The automated system marked a non-trivial number of students' correct responses incorrect, and vice versa. This course provided an authentic setting to investigate student attitudes and find better ways to incorporate imperfect autograders into the classroom.

Our first major contribution is interview and statistical evidence for what affects students' perceptions of fairness, educational value, and satisfaction in the context of an imperfect AI autograder. We show that the perceived probability of correct answers being marked incorrect (false negatives) was statistically associated with dissatisfaction and perceptions of unfairness; at the same time, participants broadly overestimated the chance of false negatives. In contrast, many participants had not considered the possibility of incorrect answers graded as correct (false positives), yet they perceived the existence of FPs as unfair and harmful to learning.

Second, we extend prior work of folk theories in social-technical systems to ASAG systems, and find that misalignment between folk theories about how the autograder worked and how it actually worked could lead to suboptimal answer construction strategies.

Third, we propose guidelines for incorporating imperfect short answer autograders into classrooms in a manner that is considerate of students' needs. Based on our findings, we propose that (1) transparency may improve students' attitudes and help students form folk theories that lead to more effective strategies for constructing answers; and (2) instructors should emphasize practice in low-stakes environments and carefully design error and attitude mitigation strategies in high-stakes environments.

2 RELATED WORK

This work sits at the intersection of multiple fields: computer science education, automated grading of natural language, and human-AI interaction. We give a brief overview of related work in each respective area.

2.1 Explain in plain English (code reading) problems

The short-answer problems that are the focus of this paper are categorized as Explain in Plain English (EiPE) problems in the computer science education literature. Note that in general, EiPE and automatic short answer grading (ASAG) are separate concepts – EiPE problems do not necessarily have to be autograded. In this paper, we use "code reading problems" to refer to the AI-autograded EiPE problems under study, because the problems asked students to read and explain code and the instructor used the term to introduce this type of problem to students.

While a full discussion of the theory and evidence behind the teaching effectiveness of EiPE problems is beyond the scope of this paper, code reading is believed to be a developmental skill that precedes code writing [38] and previous research has found that performance on EiPE problems predicts performance on code writing problems [40, 42]. In order to implement code reading problems in a class of 600 students, the instructor, a co-author of this paper, deployed a system to autograde EiPE problems.

2.2 Automatic assessment of natural language

Two major categories of automated natural language assessment stand out: automatic short answer grading (ASAG) and automatic writing evaluation (AWE). ASAG systems, like the one we investigate in this work, decide if the content in a short answer is objectively correct [10, 34]. In contrast, AWE systems focus on the writing and rhetorical quality of longer responses and essays but less so on the factual accuracy of content [2, 10]. Both types of systems are considered potential solutions for reducing the burden of grading large classes and assist student learning with instantaneous feedback [27, 55].

Both ASAG and AWE have been applied in classroom and high-stakes standardized test settings. ASAG's classroom deployments include subjects such as computer science, biology, psychology, physics, math, and reading [3, 4, 18, 31, 49, 54]. AWE has been deployed in primary, secondary, and higher education settings to help students improve their general essay-writing skills [27, 43, 51, 56]. In high-stakes settings, AWE systems appear in the standardized tests administered by states in the United States [50] and tests administered by ETS (known for the GRE and TOEFL) [11]. c-Rater, an ASAG system, was used to evaluate constructed-response math reasoning in the 2002 NAEP ICT Science test and reading comprehension items in Indiana's 2002 11th grade English End of Course Assessment [34].

2.3 Stakeholder perceptions of ASAG and AWE

Both ASAG systems and AWE systems have achieved reasonably high agreement with human graders [2, 9, 47], yet concerns still exist regarding fairness, cheating, technical issues, and validity [12, 14, 27, 50]. As a result, students' and teachers' attitudes to these systems are critical to further adoption.

There are a few studies related to students' and instructors' perceptions towards AWE systems. In a study of a middle school deployment of an AWE system, Grimes and Warschauer found mixed perceptions among students and teachers. Many students and teachers noticed errors and unreliability in the system, but at the same time teachers found benefits related to classroom management and encouraging revision [27]. Curran, Draus, and Maruschock surveyed college students and found most preferred human essay grading over computer essay grading, with somewhat more acceptance for computer-given feedback; notably, the survey did not ask about any personal experiences with any particular AWE system [13]. Roscoe et al. found that both the manner in which an AWE system was depicted before use and students' first-hand experience with the system affected their future willingness to use the system [45].

To our knowledge, there is very little work studying users' perceptions of ASAG systems. In a "no-stakes" application of a very-short-answer (four or fewer words) autograder in a medical exam, Sam et al. found positive student perceptions of the assessment's value [46]. The only other relevant work we are aware of is that of Azad et al. [5], which found that students perceived code reading questions that were graded by an ASAG system as less reliably graded than other types of exam questions; however, they did not investigate why. Given the lack of research on attitudes or presence of best practices for using ASAG, our study aims to explore this topic from students' authentic classroom experiences.

2.4 Human-AI interaction

Our work contributes to a growing body of literature about peoples' interactions with algorithmic systems. We review three major topics related to peoples' attitudes: folk theories, what affects trust in algorithms, and concerns about the fairness of algorithms.

2.4.1 Folk theories and mental models. Folk theories describe intuitive, causal explanations about a systems' functionality that develop among non-professionals based on first-hand experiences and circulate informally [20]. They can differ significantly from what is correct or accurate, but even "incorrect" theories can prove useful. For example, in Kempton's study of home thermostats [30], people used the Feedback Theory, which posits that the thermostat acts as a switch that turns on/off to maintain a target temperature, or the Valve Theory, which posits that the thermostat controls the strength or air flow of the heating/cooling system. While experts consider the Feedback Theory essentially correct, the Valve Theory could still produce advantages, such as encouraging energy savings.

Besides thermostats, researchers have described how folk theories drive interactions in algorithmic socio-technical systems. Eslami et al. discovered folk theories explained users' actions (or lack thereof) to try to coax Facebook into prioritizing desired content [20]. Another study by Eslami et al. found folk theories drove action on Yelp, where users wrote reviews to cater to how they thought Yelp's review filtering algorithm worked [21].

In addition, there is evidence that folk theories can affect people's attitudes towards the algorithms and the platforms on which they are used. After a (false) report that Twitter was going to change its feed curation algorithms, DeVito et al. found that folk theories about said algorithm framed users' tweets of dissatisfaction [16]. Ur et al.'s study of online advertising found that most participants overestimated the amount of information that trackers collected, and found privacy concerns drove lower acceptance of online behavioral advertising [52].

We expand the study of folk theories surrounding algorithmic systems to AI autograders. Similar to how folk theories inform how users cater to social media algorithms, we expect folk theories to inform strategies for constructing answers or even "gaming" the system. This, in turn has implications on how well AI grading elicits knowledge. Thus, we ask the following research questions:

- **RQ1**: What folk theories do students form around the code reading question grading process?
- **RQ2**: How do folk theories relate to students' strategies when constructing answers to code reading questions?

2.4.2 Trust. In the context of automated systems, Lee and See defined trust as "the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability" [35]. Trust in automated systems is fragile: studies by Dzindolet et al. [19] and Dietvorst et al. [17] have found that people quickly lose trust in algorithmic predictions after seeing mistakes, even when the systems outperform humans – a phenomenon known as *algorithm aversion.* Low levels of trust contributes to system disuse, but over-trust, leading to dangerous overreliance on or failure to monitor automated systems, has been observed too [19, 44]. For example, Eastern Flight 401 crashed because the crew failed to notice the autopilot was disengaged [44].

For AWE, over-trusting has prevented instructors and students from noticing inaccurate scoring [27]. On the side of low trust, Roscoe et al. found that students who were unwilling to continue using an AWE system reported lower perceived scoring accuracy (i.e. trust the system would give accurate scores) [45]. These results suggest that trust will be an important factor in the deployment of ASAG systems as well.

Finding ways to manage trust, especially with transparency, has received much attention. Merely stating a system's accuracy appears to have limited effectiveness, as Yin et al. found that trust is more tied to a users' personal experiences with a system [58]. Adding transparency into an algorithm's operation can influence trust in nonlinear and context-dependent ways. Dzindolet et al. found that giving an explanation for why errors might happen prior to use of a system increased trust, but potentially to an unwarranted level [19]. Kocielnik et al. found that when users interacted with a very inaccurate AI (that found appointments in emails), transparency improved system acceptance, but only when the system was optimized for high precision [33]. In an educational setting, Kizilcec varied transparency levels in a system that automatically compensated for peer grading bias in a high-stakes college class assignment. When students' expectations were negatively violated, either giving too little or *too much* explanation resulted in lower trust [32].

However, little work has studied ways to manage trust with imperfect ASAG systems. Forming guidelines for trust management will involve measuring trust and understanding what factors impact it. In this work, we examine accuracy perception, or the level of *confidence that the autograder will grade responses correctly*, and compare it against the actual system accuracy as a measure of trust. To explore this alignment, we ask:

- **RQ3**: What explains students' perceptions of autograding accuracy on code reading questions?

2.4.3 Fairness. A system discriminates unfairly if it systemically "denies an opportunity or a good or if it assigns an undesirable outcome to an individual or group of individuals on grounds that are unreasonable or inappropriate" [25]. Based on the literature, we succinctly summarize fairness as "judgement based only on relevant characteristics." Incidents and discoveries of algorithmic bias have been numerous [57], and ensuring fairness has been argued as a concern deserving significant attention [25], including in automatic grading systems [55].

Woodruff et al. noted that participants who learned for the first time about algorithm unfairness in products from a trusted company felt betrayal, anger, and disappointment that led to a distrust of that company [57]. In the context of education, grades significantly affect students and they care greatly about fairness in the way grades are assigned [32]. There has been work assessing whether the ratings given by AWE systems had group or individual biases compared to human graders, such as by Bridgeman et al. [8], but there is very little work assessing what biases students perceive in grading algorithms. To add knowledge to the existing body of work on fairness, we ask the following research question:

- **RQ4**: What explains students' perceptions of fairness towards code reading questions?

2.5 Other attitude measures

Our study includes three more measures of students' attitudes that intuitively relate to autograder acceptance. First, because learning is the fundamental goal of implementing code reading questions, we ask:

- **RQ5**: What explains students' perceptions of the educational/learning value of code reading questions?

Finally, we ask about students' satisfaction of and feedback for the algorithm and policies surrounding the code reading questions:

- **RQ6**: What explains students' satisfaction towards code reading questions?
- **RQ7**: What kind of feedback and desired improvements do students have for the algorithm or policies surrounding code reading questions?

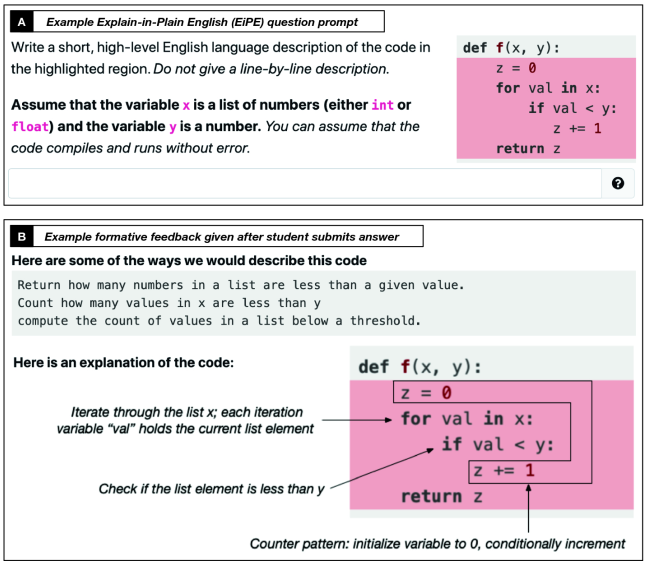
3 METHODS

First, we describe the autograder that students interacted with, and how it was presented and used. Next, we introduce the IRB-approved survey and interview that investigated students' interactions with the autograder.

3.1 Development and incorporation of the autograder

The automatic grading system that we study was developed for educational purposes and used in the Spring 2020 semester of the introductory computer science course for non-majors at our institution. Approximately 600 students enrolled in this course, which aimed to introduce basic principles of programming in both Python and Excel to people without prior programming experience.

All homework and exams were computerized and included code reading (EiPE) problems which asked students to write a short, high-level, English description of what a block of Python code achieved (Figure 1). Upon submission of an answer on a code reading problem, the system provided students with feedback on whether the answer was correct and sample exemplar solutions. This information appeared on both homework and exams. For more complicated EiPE questions later in the course, the solution also contained a visual explanation which highlighted common programming idioms and guided students in their understanding of the correct answers provided.



Example Explain-in-Plain English (EiPE) question prompt

Write a short, high-level English language description of the code in the highlighted region. *Do not give a line-by-line description.*

Assume that the variable x is a list of numbers (either int or float) and the variable y is a number. *You can assume that the code compiles and runs without error.*

```
def f(x, y):
    z = 0
    for val in x:
        if val < y:
            z += 1
    return z
```

Example formative feedback given after student submits answer

Here are some of the ways we would describe this code

```
Return how many numbers in a list are less than a given value.
Count how many values in x are less than y
compute the count of values in a list below a threshold.
```

Here is an explanation of the code:

```
def f(x, y):
    z = 0
    for val in x:
        if val < y:
            z += 1
    return z
```

Iterate through the list x; each iteration variable "val" holds the current list element

Check if the list element is less than y

Counter pattern: initialize variable to 0, conditionally increment

Figure 1: Screenshot of a code reading problem prompt (A) and feedback after a student submits an answer (B)

3.1.1 *Development and training.*

The autograder discussed here was developed iteratively from Fall 2019 onward, and is simpler than state-of-the-art ASAG. Here we discuss the most salient details of its implementation, but a full description can be found elsewhere [24]. The autograder used a logistic regression on bigram and bag-of-word features. Logistic regression was selected for its interpretability, easy customization of false positive/negative rates, and satisfactory accuracy among quickly prototyped models.

The computer preprocessed answers to minimize the impact of non-word symbols and spelling mistakes, and used a simple garbage filtering routine to discourage low-effort system gaming. Preprocessing did not apply stop word filtering, stemming, or lemmatization.

Training sets for each question consisted of manually-graded students' answers from the Fall 2019 semester deployment of code reading questions. Trained members of the instructor's research team labeled the training data in a rigorous process. For each question, the bag-of-words and bigram features were selected independently during training.

The instructor had tuned the autograder to err more on the side of false positives because they considered false negatives (i.e. correct answers graded as incorrect) more harmful to the course, resulting in a 10% FN rate and 15% FP rate. FNs were often caused by answers with conflicting words (for example, "returns whether x is odd or even" has both "odd" and "even") and valid words that were rare in the training set [24]. FPs were often caused by incorrect information that did not have negative weights, or long-distance dependencies that bigrams cannot catch, e.g., an early "not" could flip an answer's meaning [24]. Despite these imperfections, the instructor considered the error rates "good enough" for deployment because of (1) the ability to mitigate errors in high-stake settings via manual grading and appeals, (2) comparable accuracy to human TAs [24], and (3) the autograder's immediate feedback enabled students to practice EiPE questions when otherwise impossible.

3.1.2 Portrayal, use, and policies. The instructor introduced code reading problems and their grading process as a system with good accuracy but lingering imperfections. The following is an annotated transcript from the lecture in which homework-based code reading problems were introduced:

> INSTRUCTOR: Code reading questions. You have to do it a couple times. [*They bring up an example on the computer, and paraphrase the directions.*] Tell me in English what this piece of code does. [*They type and submit an obviously incorrect answer as a demonstration.*] "This makes me happy" turns out to not be the correct answer. [*They type and submit another answer.*] "It multiplies two numbers together." And that is in fact the correct answer.
>
> INSTRUCTOR: It uses NLP. Its not bulletproof, but its pretty good. *If* you encounter something you think is correct but its not grading it, please report it because we can take that example and use it to improve the algorithm.

There were some differences in the deployment of the autograder on homework versus exams. On homework assignments, students had unlimited attempts to get credit and received no penalties for incorrect submissions. On exams, students had only one attempt, as past experience showed that allowing multiple attempts resulted in an large increase of false positives but only a small decrease in false negatives [5]. The autograder on the exams still provided immediate feedback on whether an answer was correct and exemplar answers as it did on the homework.

Students could click a button that started a manual appeal process if they felt the autograder had made an error; they could also post questions and concerns on the course forum. However, an unannounced policy made appeals partially moot: course staff regraded all answers the autograder marked incorrect on exams, regardless of whether a student appealed. Staff also checked answers the autograder marked correct on exams to obtain false positive rates, but points already awarded by the autograder were never taken away.

Finally, code reading questions on the third midterm exam, the last midterm before the final exam, operated differently than all the other exams. As the models for code reading questions were not yet ready for that particular exam, the autograder informed students that their answers would be manually graded instead of graded by the autograder. When we collected data, some participants' had most recently taken the third midterm, and some the final exam. After data collection we checked for evidence of recency biases and found no statistically significant differences.

3.2 Participant recruitment

We conducted a mixed-methods (survey + interview) study about the system described above to answer our RQs. To recruit participants, two days before the final exam, a course announcement invited all students to voluntarily participate in research about "attitudes towards the assessment methods" in the course[1]. This announcement recruited for our survey and interviews simultaneously, and it instructed interested students to choose one or the other. Later, we removed any interviewees that had also taken the survey.

3.3 Survey

Students could complete the survey at any time before they received their final grades. 62 students took the survey, with a median time to completion of 570 seconds (mean=546 s and SD=211 s after removing outliers). As compensation, we invited participants to enter a lottery in which one out of every ten participants would win a $10 gift certificate. 62 students completed the survey, and we removed 13 participants (21%) who failed attention checks, leaving 49 participants' data for analysis. Of these participants, 38 responded to the survey before the final exam and 11 responded after. We designed the survey to answer RQ1 and RQs 3-6, and summarize the contents of the survey here; the supplementary materials contain the full survey.

3.3.1 Part 1: folk theories (RQ1). To gather theories about how the grader worked, we asked an open-ended question worded *"For code reading questions, what do you think happens between the time you submit your answer and the time you get feedback on that answer?"*

3.3.2 Part 2: attitudes (RQs 3-6). We surveyed the following perceptions towards the code reading questions: accuracy, fairness, educational value, and satisfaction. Note that these measures should be interpreted as a holistic view of EiPE questions, the autograder's quality, and the circumstances, portrayal, and policies associated with its use. The interviews, in contrast to the surveys, asked students to explain their answers and aimed to tease apart these factors.

We first asked about perceived fairness. Since one common definition of fairness is "judgement based only on relevant characteristics," we asked students to rate various questions' ability to "accurately reflect your knowledge of the concepts taught in the course." Students rated both code reading questions and the other types of assessments in the course for comparison: programming (i.e. code writing) problems, true/false questions, and multiple choice questions.

We had two other measures of fairness [37]. For disparate treatment between individuals, we asked for Likert agreement with "my answers for the code-reading question are graded in a similar manner as other students' answers." For disparate impact among subgroups, we asked for Likert agreement with "Some groups of students may have an advantage in regard to code reading questions." Those that had any level of agreement with that question were then asked an open-ended question about which specific groups could have a (dis)advantage.

Second, we evaluated satisfaction. Survey-takers rated their satisfaction with code-reading questions on homework and exams respectively, each using a 5-point Likert scale. Students also reported their satisfaction on the other types of course assessments to allow comparisons.

[1] We intended to release the course announcement after the final exam; however, due to miscommunication, the course staff posted it two days before the final. We found no statistically significant difference in exam satisfaction rates between those that participated before and after the final exam, suggesting a small impact if any.

Next was accuracy perception, measured with four related but distinct aspects: positive predictive value (precision), negative predictive value, false negative rate, and false positive rate. We described these aspects in plain English and asked participants to estimate their likelihoods "especially thinking about the code-reading questions on homework." The question emphasized homework assignments because students had more experience with code-reading questions on homework, especially with unlimited attempts. Possible responses were on a scale between 0% and 100%, with a step size of 10%.

Finally, we measured perceptions of educational value using a 7-point Likert agreement with "The code-reading questions helped me learn the material in the course". We focused on the educational value for the homework but not the exams because the exams were summative.

3.3.3 Part 3: explanatory factors and demographics. Finally, we measured four factors that could potentially explain variance or control for biases in the data: algorithm awareness, tech-savviness, self-reported class performance, and gender. We assessed tech-savviness with a subset of the technology anxiety instrument validated by Meuter et al [41], and then constructed a tech-savviness score by coding the Likert responses as integers and summing.

To our knowledge, a formally-validated instrument of algorithm awareness or literacy does not exist. Thus, we brainstormed seven statements about the existence of algorithms in a variety of accessible scenarios with wide-ranging implications: social media feeds, online search, online advertising, deepfakes, and automated writing evaluation. When a participant's Likert agreement with a statement (at least a slightly agree/disagree) aligned with actual algorithm existence, we termed the response "correct." The number of correct responses constituted a participant's algorithm awareness score.

3.4 Interview

All interviews took place via video conference in a two-week period after the final exam and before students received their final grades. We conducted 22 valid interviews; on average, interviews lasted 46.5 minutes. Interviewees were compensated at a rate of $20/hr.

Interviews were semi-structured. The first two co-authors each interviewed about half the participants. To ensure consistency, both interviewers conducted the first two interviews together, and every few interviews they met to discuss revisions to the interview script and to document the wordings of follow-up questions. The interview broadly followed the structure as the survey; the supplementary materials contain the full interview script.

First, we gathered theories for how the autograder worked (RQ1) and asked about students' answer construction strategies to answer RQ2. Then, we asked questions about the fairness, the four aspects of accuracy, homework/test satisfaction, and educational value (RQs 3-6). When discussing fairness, we disclosed the existence of false positives and negatives to have a richer discussion and gather additional insights for RQ1. We did not do so in the survey. This disclosure had the potential to affect accuracy perceptions most; however, we performed one-sided Welch's t-tests on accuracy perceptions between survey and interview participants and found no significant increases in false positive/negative rate perception among interviewees. During discussions of satisfaction and educational value, we additionally asked for feedback to improve these aspects (RQ7). Finally, interviewees completed an online form that contained the same measures as part 3 of the survey.

3.5 Participant characteristics

Looking at the combined data of survey and interview participants (total valid N=70), algorithm awareness and tech-savviness scores fell into normal distributions. Participants skewed towards better self-reported class performance (53% above average, 39% average, 9% below average). Gender distribution skewed towards female (70% female, 30% male, 0% other) in contrast to the overall course enrollment (46% female, 54% male). In particular, about 80% of interviewees identified as female and no interviewees reported below-average class performance.

4 RESULTS

4.1 Analysis

4.1.1 Qualitative. We had valid data for 22 interviews. For our qualitative analysis, after transcription, we employed a grounded theory-like approach [39], where we used a reflexive and iterative process to inductively generate themes that summarized interview topics. First, two members of the team performed independent thematic coding [26] for four different interviews each and then exchanged the coded interviews to review, resolving any differences with discussion. Based on these eight interviews, a number that could achieve code saturation [28], we create an initial codebook which included over 80% of the codes in the final version. Each interview question had its own set of codes.

The same two coders then resumed independent coding of the remaining interviews with the initial codebook using a deductive approach [22]. If there were statements that the initial codes could not cover, we updated existing codes or added new ones. Both coders reviewed changes to ensure necessity and accuracy. Codes achieved convergence by the end of the analysis.

Lastly, we applied a semantic approach to iteratively refine and group the codes into higher-level themes [26]. We counted the occurrence frequency of each theme to inform reporting of results. However, we did not only consider frequency; we reported on any themes that added new insights to our RQs.

4.1.2 Quantitative. We have a total of 70 participants for quantitative analysis (49 surveys + 21 interviews), since one of the 22 interviewees failed to submit their web-based questionnaire. Statistical power concerns necessitated combining survey and interview data. To make the data compatible, we converted fairness, educational value, and satisfaction measures in the survey to binary variables, and the first two authors independently extracted these binarized attitudes from the interviews with a 97% inter-rater agreement.

Statistical analysis involved six multiple regressions to explain variance among folk theories (RQ1) and the various measures of attitudes towards code reading questions (RQ3-6). The response variables for the regressions were (1) folk theory, (2) perceived false negative rate, (3) fairness perception, (4) educational value, and satisfaction on homework (5) and tests (6).

Table 1: Coefficients and p-values for the one OLS and five logistic regressions fit using combined interview and survey data (N=70). Each row contains results for one regression, and columns contain predictor variables. Note that some response variables also function as predictor variables. Cells contain coefficients and their p-values separated by a slash; blank cells indicate a predictor was not used for that regression. Highlighting and asterisks emphasize the two coefficients with p-values that fell under the B-H procedure's critical p-value of 0.012.

Regression results: coefficients / p-values							
Response variable. Logistic regression for binary (B) variables, OLS for continuous (C) variables.	**# of algo awareness Qs correct (ordinal)**	**Tech-savviness score (ordinal)**	**Self-reported class performance is avg or below (B)**	**Gender is male (B)**	**Perceived FN rate (C)**	**Perceived to be fair (B)**	**Perceived to have edu value (B)**
Had keywords-related folk theory (B)	0.123/0.593	0.086/0.237	−1.525/0.021	0.261/0.714			
Perceived FN rate (C)	−4.250/0.062	0.068/0.919	7.482/0.216	1.899/0.769			
Perceived to be fair (B)	−0.473/0.034	0.050/0.427	−0.838/0.136	0.801/0.186	−0.036/0.005*		
Perceived to have edu value (B)	−0.255/0.214	−0.110/0.072	−0.652/0.226	0.622/0.301	−0.026/0.026		
Satisfied on HW (B)	−0.390/0.073	0.033/0.587	−0.613/0.253	0.175/0.759	−0.015/0.207	0.116/0.855	0.380/0.540
Satisfied on tests (B)	−0.572/0.034	0.029/0.706	−0.393/0.577	1.108/0.104	−0.041/0.001*	0.060/0.944	1.381/0.113

We chose to only predict false negative rate since (1) perceived precision and false positive rates had little variation; (2) interviewees had an easier time reasoning about and remembering false negatives than negative predictive value; and (3) the Pearson correlation between negative predictive value and false negative rate was -0.504 ($p < 0.001$), suggesting similar information content of the two metrics.

Each regression's predictors included the participant characteristics gathered from part 3 of surveys and interviews. In addition, we used other dependent measures as predictors where we *a priori* thought associations might reasonably occur; for example, we suspected perceived false negative rate may affect satisfaction. Table 1 summarizes these hypotheses, along with associations found to be significant (we detail the process for finding significance below).

The six regressions fit a total of 32 coefficients, not including intercepts. Due to the risk of Type I errors, we used the Benjamini-Hochberg procedure [7] to control the false discovery rate, or the expected proportion of significant results that are false positives. The B-H procedure outputs a critical p-value, where all observed p-values under the critical p-value are considered statistically significant.

Given the exploratory nature of this study and limited data, we chose a false discovery rate of Q=0.20 to provide suggestions for future investigation and not to definitively identify any associations or their strength. Furthermore, we support significant associations with interview evidence. Two statistically significant regression coefficients fell under the critical p-value of p=0.012.

4.2 Folk theories (RQ1)

4.2.1 Overall grading process. We gathered 22 folk theory responses in the interview and 44 valid responses from the survey, after removing four survey participants who said the grading was manually done and one that wrote they had no feedback. Five survey participants said the autograder was an automatic process, providing no additional details. The rest of the participants shared the same theory of *automated matching to exemplar answers or words*. We named three variations of this theory, each providing more detail than the last:

- **General Matching**: this theory stated that the autograder compared students' responses to the correct response(s), without specifics on the exact similarity metric or its computation. A total of nine survey participants (20.45%) and one interviewee (4.55%) reported this theory.
- **Keyword Matching**: a conjecture that the algorithm checked if a set of important words existed in an answer, but did not provide additional detail about the process. There were 29 survey participants (65.91%) and eight interviewees (36.36%) with this theory.
- **Keyword Matching with Details**: same as the Keyword Matching theory, but with at least one additional detail, which we list below. 13 interviewees (59.10%) and no survey participants used this theory, possibly because open-ended survey responses tend to be less detailed in general.

Possible details for the Keyword Matching with Details theory based on the interview responses included:

- **Needs enough matches**: Seven participants (31.82%) mentioned an answer was graded as correct only when "enough of your words match," or the autograder used "a certain cutoff percentage" to determine if enough of the words match the answer key. The autograder's logistic regression indeed used a decision threshold.
- **Covers a range of expressions**: Seven participants (31.82%) suggested that the autograder accepted a variety of ways to frame an answer, such as using words "in different order[s]" or with "different connectives," but four of them doubted it could cover all possible answers.

- **Matches sequence of words**: Three participants (13.64%) thought the autograder matched "certain phrases" or words "in certain orders" in addition to or instead of individual words, which aligned with the bigram features the algorithm actually used.
- **Uses machine learning**: Three participants (13.64%) used the phrase "machine learning."
- **Uses past answers**: One participant (4.55%) mentioned the autograder "compared it [students' answers] to past answers," which aligned with the actual development process that used answers from past students.
- **Penalizes unwanted expressions**: One participant (4.55%) articulated the possibility that the algorithm penalized certain expressions because "that's not what they want." The autograder indeed assigned negative weights to certain words and phrases.

Over 75% of all the participants held a keyword-related theory. Few participants' responses contained the more nuanced concepts, such as the use of bigrams (sequence of words), past answers, and negative weights (unwanted expressions).

4.2.2 False negatives and positives. Interviewees had various explanations for the presence of false negatives and positives. Since all but one interviewee used a keywords-related theory, these explanations tended to be framed with that in mind.

The most popular explanation for false negatives (correct answers marked incorrect), used by eight students (36.36%), stated the autograder did not or could not exhaustively cover all correct keywords, expressions, or perspectives:

> *I feel like maybe the people who designed the answers to that question might not have actually thought from a different perspective that a student might have. And then, those keywords might have been left out.* (P-ID=32)

But these explanations almost always failed to consider the large number of past answers that comprised the training data. However, there was some merit to this explanation, according to the instructor's research team: uncommon wordings such as "largest variable" instead of "largest number" could indeed cause false negatives.

Other explanations for false negatives that each comprised less than 20% of participants included too "strict" of a decision threshold and misspelled words. Lastly, two participants (9.10%) blamed technical issues or bugs and four (18.18%) had no cohesive explanation.

As for false positives (wrong answers graded as correct), four participants (18.18%) could not think of any reasons at all, and two (9.10%) blamed technical issues. In the remaining interviewees, 11 (50%) participants reasoned that answers with the correct keywords but the wrong meaning as a whole caused false positives. Some of them provided more details, such as keywords in an incorrect order (N=2, 9.10%), or answers containing both correct and incorrect information (N=1, 4.55%). The instructor confirmed these possibilities.

The above theories imply some interpretability contained in responses, but some participants (N=3, 13.64%) suspected that the system could grade agglomerations of keywords incoherent to humans as correct, i.e. cheating or gaming the system. When we asked participants about the prevalence of system gaming, 15 (68.18%) expressed it would likely involve at least the same amount of effort as responding to the questions legitimately, and no one reported instances of successful gaming.

4.2.3 Folk theory formation. Around 60% (N=13) of the interviewees stated their theories originated from first-hand experience with the autograder's grading behavior and by comparing their submitted answers to the sample exemplar answers contained in the feedback. Exemplar answers could function as key signals of what styles of answers to write and *not* to write, especially when students thought they experienced false negatives (N=4, 18.18%). And, participants (N=4, 18.18%) explained how they identified patterns from exemplar answers to inform the Keyword Matching theory:

> *I think that in the example that they give to you after you finish answering the question in the practice, most of them contain the same few like keywords that indicate what this code is trying to accomplish.* (P-ID=35)

Four interviewees (18.18%) used logical reasoning to arrive at a keyword-related theory. The reasoning process could be quite different across students. For example, one student reasoned that the large variety of possible answers required the autograder's designers to extract what was common to the answers: the keywords. Another student started with the opposite premise, arguing that only a limited number of answers existed for any basic coding idea, and so any correct response must share some of the keywords with the answer keys.

Finally, one participant (4.55%) mentioned that they used instructor statements and course information to inform their theory. For this participant, the instructor's mention of NLP during lecture confirmed their suspicion that the autograder relied on machine learning. For our other participants, it is possible that they did not know what "NLP" meant, which reduced the influence of the instructor's statements.

4.3 Answer construction strategies (RQ2)

In this section, we discuss strategies our interview participants used to construct answers to code reading problems. Participants generally formed and refined answer construction strategies for code reading questions via trial and error over the semester concurrently with folk theory formation. 13 interviewees (59.10%) said they learned to cater their answers to the autograder by imitating the exemplar answers from the feedback and finding patterns in what kinds of answers worked. More specifically, nine participants (40.91%) said they had learned the patterns for translating common programming paradigms and control structures from code into plain English patterns.

The Keyword Matching theory strongly informed participants' strategies. Half of the participants (N=11, 50%) thought the autograder preferred technical terms learned in class over vernacular – for instance, preferring "concatenate" over "combine words." Often times participants thought the autograder looked for a rather limited set of keywords, so they tried to imitate the writing style and sentence structure of the exemplar answers to increase the perceived chance that the algorithm would be looking for their words. A few participants (N=4, 18.18%) tried to memorize the exemplar answers or the keywords on homework problems so that they could get those problems correct if they encountered the same or similar questions again.

One interviewee's folk theory led to a suboptimal strategy. This student thought the autograder used a threshold, and reported writing long answers to capture as many keywords as possible to pass the threshold. However, the student was not aware that the grading algorithm weighed some words negatively and that longer answers correlated with lower autograder scores in most cases. As a result, this strategy could have harmed the student by introducing unwanted keywords into their answers. The student noted that their strategy was only sometimes successful.

In summary, participants formed their answering strategy gradually, often by leveraging their folk theories and learning from the exemplar answers. Misalignment between folk theories about how the autograder worked and how it actually worked could lead to suboptimal answer construction strategies.

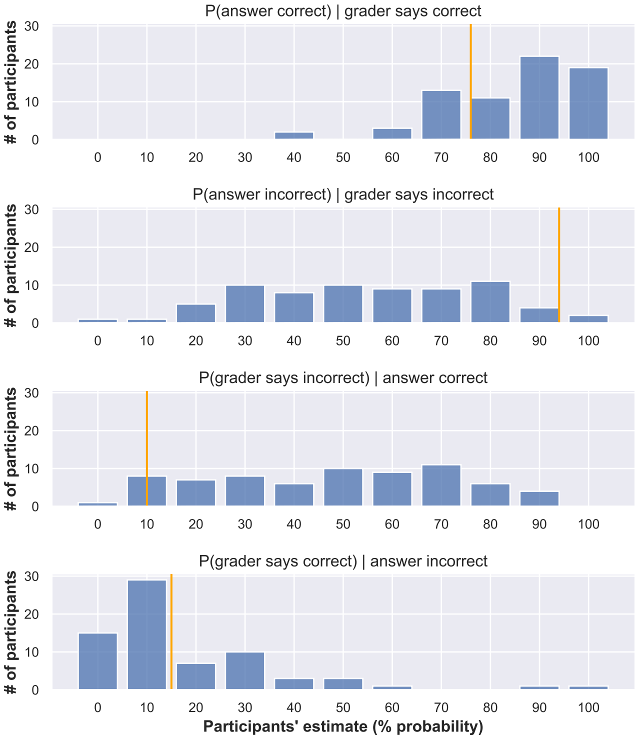


Figure 2: Students' accuracy perceptions of the autograder, with respect to positive predictive value, negative predictive value, false negative rate and false positive rate from top to bottom. It shows data from both the survey and interviews (combined N=70). The system's actual performance is annotated with an orange line, located at 76%, 94%, 10%, and 15% respectively. In contrast, means of student estimates were 85%, 55%, 48%, and 17% respectively.

4.4 Perceptions of accuracy (RQ3)

Figure 2 shows histograms of survey and interview participants' accuracy perceptions of the autograder, with the system's actual performance annotated with an orange line. About 30% (N=15) of the participants responded with a false positive rate of zero, implying they thought false positives were nonexistent or near-nonexistent. On the other hand, participants tended to overestimate the chances that correct answers would be marked wrong (false negatives) – as the middle two plots show, most were quite far from the orange line. Contrary to how the autograder actually performed, 90% of the students thought false negatives happened more often than false positives.

Most interviewees (N=19, 86.30%) said they had not considered the possibility of false positives at all. One participant reported having personal experience with false positives, and no one had heard of it happening from others. A few interviewees (N=3, 13.64%) said they estimated a 10% false positive rate because we told them the existence of false positives, but as discussed before, this did not result in noticeably different false positive estimates compared to survey participants.

One of the most common reasons for interviewees to estimate a high false negative rate, mentioned by seven interviewees (31.82%), was they thought they had multiple encounters with false negatives. Another common influence was hearing about false negatives from other students in the course frequently (N=7, 31.82%).

In summary, students tended to overestimate false negative rates, and many underestimated false positive rates. Interviewees mentioned first-hand experience and what they heard from other students as key factors influencing their estimates.

4.5 Perceptions of fairness (RQ4)

Figure 3 summarizes the survey data from our primary measure of fairness, based on the definition of "judgement based only on relevant characteristics," Survey takers on average thought that programming questions and multiple choice questions were fairer than code reading questions, and fairness perceptions for code reading questions were polarized. Logistic regression on the polarity of fairness perception using the combined survey and interview data found participants with higher false negative rate perception were statistically significantly more likely to say the autograder and its surrounding policies were unfair (p=0.005, coeff=-0.036, 95% CI=[-0.062, -0.012]).

We now turn to the interviews for more in-depth insights. False positives/negatives and the course's appeal system emerged as major drivers of perceptions of overall fairness. Around 70% of the interviewees (N=15) expressed discontent towards false negatives (correct answers graded as wrong), while five of them (22.73%) explicitly mentioned them in why they thought the code reading questions inaccurately assessed their course-related knowledge. The course's appeal policy mitigated these concerns to some extent: six participants (27.27%) expressed that false negatives frustrated and inconvenienced them, but the appeal process could ultimately resolve the errors. Another seven participants (31.82%) had neutral feelings towards false negatives, either because of the appeal process, because they expected errors to happen in large courses, or because they only experienced them in a low-stake setting.

When we elicited views about false positives, 13 participants (59.10%) expressed fairness concerns about non-knowledgeable students getting credit that they did not deserve. Unlike false negatives, which appeals could mitigate, participants could not think

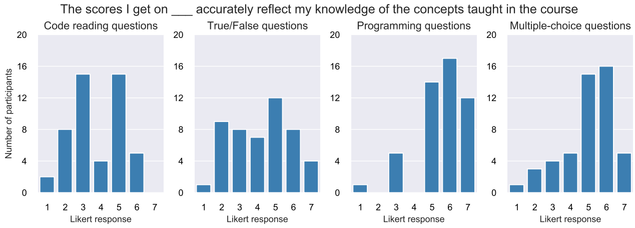


Figure 3: Distribution of fairness ratings for each type of assessment method in the course, measured by rating agreement with the statement that "The scores I get on ___ accurately reflect my knowledge of the concepts taught in the course" where the blank contained "code reading questions", "true/false questions," "programming questions," and "multiple choice questions" respectively. Ratings were based on a 7-point Likert scale – 1: Strongly disagree, 2: Disagree, 3: Somewhat disagree, 4: Neither agree nor disagree, 5: Somewhat agree, 6: Agree, 7: Strongly agree. Survey data only (N=49). Survey takers on average thought that programming questions and multiple choice questions were fairer than code reading questions, and fairness perceptions for code reading questions were polarized.

of solutions to correct false positives, and they recognized that students had less incentive to report false positives (N=7, 31.82%). Besides this, six interviewees (27.27%) thought false positives could mislead students, thus impacting the educational value of code reading questions.

We additionally asked on the survey and interview for subgroups of students that could be at an unfair dis(advantage) for code reading questions. About 50% of the survey takers and all interviewees named at least one such subgroup. Participants felt concern towards non-native English speakers most frequently (N=18, 25.3%); they felt that non-native speakers could be worse at constructing answers in English or finding the right language to cater to the autograder. Groups with advantages that participants named included those with prior experience with short answer autograders (N=2, 2.82%), and students with prior experience in computer science (N=11, 15.49%), though many participants considered this advantage fair. For our other measure of fairness, disparate treatment between individuals, five out of 49 survey respondents disagreed that all students' answers were graded in a similar manner, showing that students were less concerned about disparate treatment than disparate impact.

In summary, participants were polarized on whether they thought the autograder and its use was fair. False negatives and positives impacted interviewees' fairness perceptions, but the course's appeal policy helped mitigate concerns about false negatives somewhat. Many students named a subgroup of students that they thought could possibly experience an unfair (dis)advantage.

4.6 Perception of educational value (RQ5)

We asked one question regarding educational value of code reading questions in the survey and the interview. Figure 4 summarizes the survey responses; about half of survey participants agreed to any extent that the code reading questions helped them learn the material in the course.

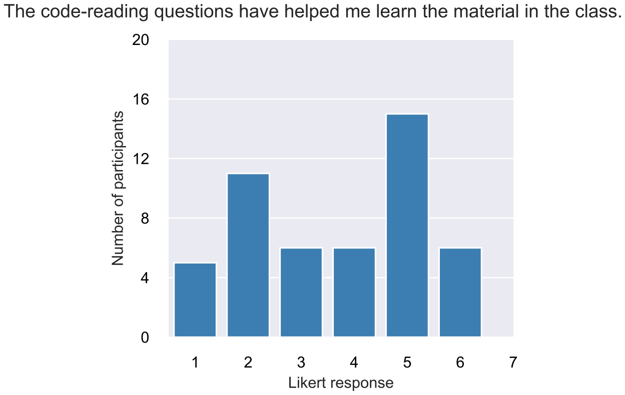


Figure 4: Distribution of perception of educational value surfaced by agreement with the statement that "The code reading questions have helped me learn the material in the class" on a 7-point Likert scale – 1: Strongly disagree, 2: Disagree, 3: Somewhat disagree, 4: Neither agree nor disagree, 5: Somewhat agree, 6: Agree, 7: Strongly agree. Survey data only (N=49). Perception of educational value was polarized.

About half of the interviewees said unequivocally that code reading questions helped them learn, reporting that the questions helped them identify and fix weaknesses in their understanding of course concepts, and that they recognized the value of being able to read code and communicate its function. In contrast, ~25% of interviewees said code reading questions did help learning but also surfaced concerns, and the remaining ~25% of interviewees considered code reading questions unhelpful for learning.

Interviewees mentioned several factors that negatively impacted their perceptions of educational value: false negatives/positives, the reliance on memorizing answers, and insufficient instruction (N = 4, 2, and 1 respectively; 18.18%, 9.10%, and 4.55% respectively).

Besides educational value, we asked the interviewees about their willingness to use code reading questions with the same course policies if they hypothetically were to take the class again. Nine interviewees (40.91%) gave an affirmative answer, three of them (13.64%) said yes on homework but not on exams, another three (13.64%) were willing if the autograder were improved, and the remaining six (27.25%) answered in the negative. Positive perceived educational value was necessary for students to respond in the affirmative, but it was not sufficient. For this question, participants' concerns about fairness, inaccuracy, and peer discontent sometimes outweighed positive educational value.

4.7 Satisfaction (RQ6)

We measured students' satisfaction with code reading questions in low-stakes scenarios (homework) and high-stakes scenarios (exams) separately. Figure 5 summarizes Likert ratings from the survey of each type of assessment method in the course. Three observations summarize the data:

- Satisfaction ratings of code reading questions were polarized for both homework and tests.
- About the same number of students were satisfied with code reading questions on homework as there were dissatisfied, but more students were dissatisfied with code reading questions on tests.
- Overall, students were more satisfied with the other types of assessment.

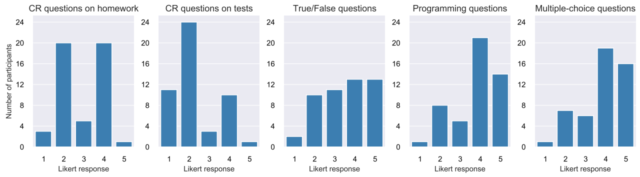


Figure 5: Distribution of satisfaction ratings for each type of assessment method in the course on a 5-point Likert scale – 1: Very dissatisfied, 2: Somewhat dissatisfied, 3: Neither satisfied nor dissatisfied, 4: Somewhat satisfied, 5: Very satisfied. "CR" is shorthand for "code reading." Survey data only (N=49). Satisfaction ratings of code reading questions were polarized for both homework and tests. More students were dissatisfied with code reading questions on tests. Overall, students were more satisfied with the other types of assessment.

The interviews revealed a variety of reasons for both satisfaction and dissatisfaction. For code reading questions on the homework, six participants (27.27%) liked the policy of unlimited attempts on the homework because it offered them abundant practice opportunities. In addition, seven participants (31.82%) gave positive feedback about the immediate feedback and exemplar answers; they said the feedback helped them find out why they were wrong and learn to cater to the algorithm for the exams. Four of the interviewees (18.18%) said they were satisfied with the code reading questions on homework because they perceived high educational value.

On the other hand, frequently being marked wrong and/or false negatives led to dissatisfaction on the homework. Two participants (9.10%) disliked code reading questions mainly because they did not perform well on them. The remaining interviewees (N=7, 31.82%) attributed dissatisfaction to false negatives. They disliked spending extra effort on the homework to learn how to cater to the system, to observe patterns, and to memorize the exemplar answers when they felt they had already understood the key concepts behind code reading.

> *I kind of felt like it was just a guessing game. ... even if I did understand the concept that they were testing ... and so, my strategy was to kind of, to write whatever I was thinking, and then, analyze the examples that they gave me, and then go through the question multiple times, to ensure that I kind of knew what they were looking for ... I definitely found myself trying to memorize a lot of the answers, so that on the test, I could simply write it out and kind of just get that question over with.* (P-ID=35)

Two themes in interviewee statements explained why participants (both interview and survey) were generally less satisfied with code reading questions on exams. First, the high-stakes nature of exams likely amplified the effects of false negatives. Five interviewees (22.73%) reported experiences of false negatives on the homework, but they felt little impact. In contrast, six interviewees (27.27%) reported that false negatives on exams frustrated, annoyed or stressed them. Indeed, logistic regression using the combined survey and interview data found that higher perceived false negative rate predicted dissatisfaction on exams (p=0.01, coeff=-0.041, 95% CI=[-0.0754, -0.0121]), but not on homework. Second, students only had one attempt on code reading questions on the exams, with no partial credit. Five participants (22.73%) felt that compared to the homework, this policy deprived them of a feeling of control over their grades.

Lastly, students stated various reasons for why they were more satisfied with the other types of questions, i.e. multiple-choice, true-false and programming (code writing). Seven participants (31.82%) preferred the other types of questions because they considered the grading more objective, with near-perfect or perfect accuracy. Seven participants (31.82%) preferred programming questions due to higher perceived educational value, or the view that code writing eclipsed code reading in relevance.

In summary, many factors contributed to satisfaction, including the autograder's errors, grading policy surrounding the autograder, the autograder's feedback, students' performance on code reading problems, and perceived educational value. Participants viewed unlimited homework attempts and immediate feedback positively, but cited the need to learn to write for the autograder, having only a single attempt on exams, false negatives, and being marked wrong as major drivers of dissatisfaction.

4.8 Feedback and desired improvements (RQ7)

At relevant points in the interviews, we asked interviewees for feedback on the code reading questions and what changes would improve their attitudes. Nine interviewees (40.91%) suggested the need for formal instruction on how to write for the autograder, as they felt the course relied on students to form their own strategies. Instruction could include explanations of the algorithm's operation, writing tips, and worked examples in lecture and discussion sections:

> *In lecture or lab, they don't really talk about those, so maybe just doing a couple of practice ones. And I had taught myself how the computer thinks about them, but maybe explaining it and showing us would be a little more helpful, and people wouldn't, I guess, get them wrong as much even though they were right.* (P-ID=2)

Related to the theme of more instruction, nine participants (40.91%) wanted more tips for explain in plain English questions in general, without specifically mentioning the autograder. They asked for more instructional activities (e.g. a dedicated lecture or explaining code to peers in discussion sections) to practice and learn before encountering the questions for real credit.

The most popular requests (N=17, 77.3%) involved improving the autograder's accuracy by various means. Six students (27.27%)

suggested collecting more exemplar answers to improve the autograder's coverage of keywords. Some proposed double-checking for autograder mistakes, especially in the exams (notably, this already was happening). Others proposed a continuation of assessments based on reading code, but switching from a open-ended response scheme to a multiple-choice or fill-in-the-blank scheme. Finally, nine participants (40.9%) preferred manual grading, because they felt it was difficult for computers to interpret natural language well. But contrary to this belief, the autograder made about the same number of mistakes as average-experienced TA graders and made fewer mistakes than inexperienced TAs [24].

5 DISCUSSION

In this section, we discuss the concerns our participants raised and what strategies might address these concerns. We conclude with a discussion of how to determine whether educators should use AI autograders.

5.1 The overestimation of false negatives

As mentioned in Section 4.4, participants broadly overestimated false negative rates (FNRs), i.e., correct answers graded as wrong, and estimates of this rate were negatively associated with satisfaction on exams and perceptions of fairness (Table 1). These associations require further research to confirm and find causation. Nonetheless, the quantitative and interview evidence of the association between FNR perception and student attitudes suggests the need for better alignment of FNR perceptions with actual system performance. Based on our interviews and prior literature, we propose three explanations and potential solutions for the FNR overestimation. Note that these explanations are not mutually exclusive nor exhaustive.

Our first explanation states that students were more sensitive to false negatives or remembered them better, i.e., the well-documented phenomenon of **algorithm aversion** [17], where people quickly lose trust in algorithmic predictions after seeing mistakes, even when the systems outperform humans. Setting proper expectations can help counter algorithm aversion, and providing transparency around algorithmic processes before use has shown promise in the literature [19, 33]. One avenue for transparency for our case is to inform students of the training process which used past students' answers, as all but one interviewee failed to mention this in their folk theories. Informing students of such components of the process signals the diversity of "correct" response expressions that the autograder accepts, to help students align their FNR perceptions with the actual FNR. Another suggestion is for the interface to highlight spelling mistakes and then give an opportunity for corrections before grading. This approach directly addresses interviewees' concerns about misspellings causing false negatives.

Second, **other students' complaints** may explain high FNR perception: a number of interviewees cited the large number of complaints on the course forum to support their estimate of false negatives (Section 4.4). We expect that mostly students with negative experiences wrote posts, leading to biased impressions. To address this, we suggest instructors first encourage students to appeal via a dedicated, private channel, and address public complaints promptly, especially those that were true negatives mistaken as false negatives.

A third plausible explanation is that students **cannot accurately differentiate between true and false negatives**. That some interviewees reported personally experiencing false negatives "frequently," despite the system's 10% FNR, lends support to this explanation. Moreover, Azad et al. found that the ways students appealed autograder decisions suggested that they did not reliably detect mis-scoring [5]. If this explanation proves to be influential, instructors should design measures to help students better self-reflect and detect mistakes. The system in our study disclosed correct answers. Beyond this, we suggest adding explanations for why common incorrect answers are incorrect (perhaps by highlighting words with high negative weights), which will help students be less quick to blame the autograder when their answers are marked incorrect.

5.2 Managing the impact of false positives

Despite the instructor intentionally biasing the algorithm towards false positives (incorrect answers graded as correct), many of our participants were not aware that false positives were possible and underestimated their frequency even after we revealed their existence (Section 4.4). The presence of false positives, however, had noticeable impact: interviewees expressed concerns about fairness and of leading students to believe that incorrect concepts were correct, thereby inhibiting learning (Section 4.5).

To address concerns about teaching misconceptions to students, we recommend instructors explore mechanisms that encourage reflection by nudging students to compare their answers to the correct answers as opposed to merely showing correct answers. Articulating the existence of false positives is another approach, however it may actualize the fairness concerns of our participants for students that had not considered the possibility of false positives before.

Fairness concerns associated with false positives are challenging to address. For low-stakes settings, future work should examine whether having low point values or emphasizing the formative aspects of the assignments will lessen concerns about undeserved points. Another mitigation strategy that we suggest for high-stakes settings is to have humans in the loop to garner trust – perhaps peer confirmation. Furthermore, technical advancements might steadily push false positive rates towards zero, but further work will be required to determine what rate students will accept.

5.3 Balancing false negatives and positives

As discussed above, both FPs and FNs cause harm. In general, system designers may chose to bias errors towards more FPs or more FNs, but as far as we know, no research has extensively studied the effects of different emphases of FPs and FNs for autograders. Future experiments are necessary to measure the causal effect sizes between FN rate/FP rate, fairness perception, satisfaction, and learning outcomes. Pedagogically, we advise that learning outcomes take priority. We do not advocate lowering FNs to increase student satisfaction at the cost of more FPs if it adversely impacts learning. In addition, since perceived FN rates were more negatively associated with exam satisfaction than homework satisfaction, this suggests that different stakes may call for different balancing strategies. For example, decreasing FPs for formative situations like homework may be justified if research shows that FPs tend to teach misconceptions, but this would not apply to high-stakes summative exams if FNs are shown to greatly decrease satisfaction.

5.4 Course policy and use suggestions for incorporating AI autograders

5.4.1 Help students construct effective folk theories for the autograder. Students in our study felt they received minimal information about how the autograder worked and instructions on how to write for it (Section 3.1.2), and some interviewees requested more explanation or examples on how to cater to the autograder (Section 4.8). While pedagogically we struggle with the notion of teaching students how to satisfy a particular autograder, our study suggests that insufficient understanding of the algorithm is problematic. Some interviewees resorted to memorization of answer keys to make the algorithm consistently award points, and at least one interviewee's folk theory about how the autograder worked did not consider negative weights, which resulted in a counterproductive strategy that included writing long answers (Section 4.3).

We feel that additional transparency would have benefited our interviewees by guiding them towards folk theories that better informed their strategies and answers. The knowledge that keywords had negative weights would discourage indiscriminately packing keywords into answers, and the knowledge that the algorithm considered groups of words would encourage students to write complete sentences rather than engage in keyword bingo. Similarly, knowing that the autograder was trained using previous students' answers in the course might help students fixate less on the exact wording of their answers.

How to provide this transparency is an open question. Messaging is one approach. Instructors should strive to find proper messaging to dispel unproductive folk theories and improve attitudes. Poor messaging can backfire: prior literature suggests that too much transparency can lead to a drop in trust [32]. Future work should explore how the amount, kind, and timing of information revealed about the autograder affects folk theories and perceptions of the system.

To better inform efforts to dispel harmful folk theories, future work should explore the efficiency of various approaches in probing students' potentially harmful folk theories; some of these approaches include surveys and analysis of discussion forums. As a bonus, these approaches will help detect gaming attempts. Our interview findings suggest that making the system hard to game is the best way to prevent gaming.

Finally, instructors should not neglect to teach general strategies for answering questions, regardless of the presence of an autograder. The goal is to teach code understanding, *not* how to write for an autograder. Many interviewees requested more instruction and practice around how to code-read and communicate a piece of code before applying the skills in an autograded environment (Section 4.8).

5.4.2 Encourage a practice mindset in low-stakes environments. Participants in our study were generally more satisfied with the autograder's use and policies on homework than in exams, and appreciated the feedback and allowance of unlimited attempts (Section 4.7) – thus, the homework effectively operated as a sandbox environment for students. Students said they formed their folk theories and answer construction strategies gradually through low-stakes practice before utilizing them on the exams, further increasing the usefulness of a good implementation of low-stakes practice. Given these findings, instructors should consider autograder use in low-stakes, sandbox-like environments to offer students practice – more of a learning than an assessment tool.

5.4.3 Address concerns about group disadvantages. Many of our interviewees mentioned certain groups experiencing unfair disadvantages, such as non-native English speakers. We do not have data on whether this was the case, but believe instructors should attempt to find out and preemptively address or dispel such student concerns if possible.

5.4.4 Approach high-stakes usage with caution. According to some interviewees, the policy of only one attempt with an imperfect autograder and all-or-nothing grades dissatisfied them and deprived them of a sense of control (Section 4.7). A system of partial credit could address some of this concern. However, offering multiple attempts as in the homework could backfire, as Azad et al. found that multiple attempts on a similar autograder significantly increased FPR but only slightly reduced FNR [5].

In addition, we recommend a robust appeal process. Interviewees said the appeal process reduced discontent towards false negatives, as shown in Section 4.5. However, the additional effort required to go through the process did cause frustration, and as suggested before, students may not know when to appeal [5]. Appeals invite fairness concerns too, as some participants had the impression that some of their peers appealed often, even when the appeals were not warranted, while others experienced self-doubt and chose not to appeal. Future work is required to address these concerns.

5.5 When is an AI autograder appropriate?

The above discussion focused on *how* to deploy an AI autograder, but before that happens, we encourage educators to ask "*should* I deploy an AI autograder?" Our findings suggest that autograder performance, subgroup fairness, and consequences of scores are at best the minimum set of criteria to consider [55]. Even if the autograder in our study had *exceeded* human performance, it is likely that algorithm aversion will still exist, students will still overestimate error rates, and they will have the same fairness concerns. Educators should carefully consider the ability and resources to address these concerns. Moreover, there exist ethical concerns as to whether an algorithm should be used as a predictive grading system. Unless we ask ourselves whether an algorithm can reliably, consistently, and ethically assess and predict outcomes – and critically address this question, we will see more cases like the recent use of an algorithm that predict A-level exam scores (for exams that were not taken) in the UK, which prompted outrage and accusations of unfairness [6].

6 LIMITATIONS

While authentic classroom experiences frame our study, the findings of our research reflect the particular autograder under study, EiPE questions, and the course policies. Thus we urge consideration of similarity to our circumstances before generalizing our findings and guidelines to other types of autograders, question types, subject matters, or student populations.

Another limitation is that interviewees could have rationalized their attitudes, creating reasons to explain their feelings rather than reporting the true explanation. Future work can apply non-self-reporting based approaches or randomized controlled experiments to expand on our results.

We note several sources of potential bias. First, our participants are subject to self-selection bias. We noted an over-representation of female participants and very few participants that reported below-average class performance, especially on the interviews. Second, surveys and interviews often suffer from self-reporting bias [1], including recall bias and recency bias. However, we found no statistically significant difference in exam satisfaction rates between those that participated before and after the final exam, suggesting a small impact. Third, a researcher's position and personal bias may impact data qualitative collection and analysis [15]. Members of the course staff and the instructors are co-authors of this paper; however, they did not actively participate in the data collection and analysis process. Those who did data collection and/or analysis tried their best to hold neutral and open attitudes towards the autograder.

7 CONCLUSION

Through our exploratory mixed-methods study, we identified factors that affect students' attitudes and interactions with imperfect AI autograders, as well as guidelines for incorporating imperfect short answer autograders into classrooms in a manner that is considerate of students' needs. Much work remains to fully capture the complexity of students' views surrounding automatic assessment. We encourage further research on solutions for the concerns of stakeholders, paving the way for AI to provide further gains in efficiency and learning.

ACKNOWLEDGMENTS

A big thanks to the students that participated in our study, the course staff that assisted us in recruiting students, and the members of the instructor's research team (Sushmita Azad, Binglin Chen, and Matthew West) who gave feedback on methodology and answered many questions about how the course worked. Additional thanks to Yiran Hou, Wenxuan "Wendy" Shi, Vinay Koshay, and Beleicia Bolluck for their writing suggestions and feedback. This work was supported by NSF grant CHS-1564041 and Capital One.

REFERENCES

[1] Alaa Althubaiti. 2016. Information bias in health research: definition, pitfalls, and adjustment methods. *Journal of multidisciplinary healthcare* 9 (2016), 211.

[2] Yigal Attali and Jill Burstein. 2004. Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series* 2004, 2 (2004), i–21.

[3] Yigal Attali and Don Powers. 2008. Effect of immediate feedback and revision on psychometric properties of open-ended GRE® subject test items. *ETS Research Report Series* 2008, 1 (2008), i–23.

[4] Yigal Attali, Don Powers, Marshall Freedman, Marissa Harrison, and Susan Obetz. 2008. AUTOMATED SCORING OF SHORT-ANSWER OPEN-ENDED GRE® SUBJECT TEST ITEMS. *ETS Research Report Series* 2008, 1 (2008), i–22.

[5] Sushmita Azad, Binglin Chen, Maxwell Fowler, Matthew West, and Craig Zilles. 2020. Strategies for Deploying Unreliable AI Graders in High-Transparency High-Stakes Exams. In *International Conference on Artificial Intelligence in Education*. Springer, Springer International Publishing, Cham, 16–28.

[6] BBC. 2020. A-levels and GCSEs: How did the exam algorithm work? https://www.bbc.com/news/explainers-53807730. Accessed 2020-09-16.

[7] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57, 1 (1995), 289–300.

[8] Brent Bridgeman, Catherine Trapani, and Yigal Attali. 2009. Considering Fairness and Validity in Evaluating Automated Scoring. In *Annual meeting of the National Council on Measurement in Education* (San Diego, CA, USA). NCME, Mt. Royal, NJ, 1–18.

[9] Brent Bridgeman, Catherine Trapani, and Yigal Attali. 2012. Comparison of human and machine scoring of essays: Differences by gender, ethnicity, and country. *Applied Measurement in Education* 25, 1 (2012), 27–40.

[10] Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education* 25, 1 (2015), 60–117.

[11] Jill Burstein, Joel Tetreault, and Nitin Madnani. 2013. The e-rater automated essay scoring system. In *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, Mark D Shermis and Jill Burstein (Eds.). Routledge, New York, NY, USA, Chapter 4, 55–67.

[12] Monica Chin. 2020. These students figured out their tests were graded by AI — and the easy way to cheat. https://www.theverge.com/2020/9/2/21419012/edgenuity-online-class-ai-grading-keyword-mashing-students-school-cheating-algorithm-glitch. Accessed 2020-09-13.

[13] Michael J Curran, Peter Draus, George Maruschock, and T Maier. 2013. Student perceptions of project essay grade (PEG) software. *Issues in Information Systems* 14, 1 (2013), 89–98.

[14] Paul Deane. 2013. On the relation between automated essay scoring and modern views of the writing construct. *Assessing Writing* 18, 1 (2013), 7 – 24. https://doi.org/10.1016/j.asw.2012.10.002 Automated Assessment of Writing.

[15] Norman K Denzin and Yvonna S Lincoln. 2008. Introduction: The discipline and practice of qualitative research. In *Strategies of qualitative inquiry* (3 ed.). Vol. 2. Sage Publications, Inc, Thousand Oaks, CA, USA, Chapter 1, 1-–43.

[16] Michael A. DeVito, Darren Gergle, and Jeremy Birnholtz. 2017. "Algorithms Ruin Everything": #RIPTwitter, Folk Theories, and Resistance to Algorithmic Change in Social Media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3163–3174. https://doi.org/10.1145/3025453.3025659

[17] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. 2015. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General* 144, 1 (2015), 114.

[18] Myroslava O Dzikovska, Johanna D Moore, Natalie Steinhauser, Gwendolyn Campbell, Elaine Farrow, and Charles B Callaway. 2010. Beetle II: a system for tutoring and computational linguistics experimentation. In *Proceedings of the ACL 2010 System Demonstrations*. Association for Computational Linguistics, Stroudsburg, PA, USA, 13–18.

[19] Mary T. Dzindolet, Scott A. Peterson, Regina A. Pomranky, Linda G. Pierce, and Hall P. Beck. 2003. The role of trust in automation reliance. *International Journal of Human-Computer Studies* 58, 6 (2003), 697 – 718. https://doi.org/10.1016/S1071-5819(03)00038-7 Trust and Technology.

[20] Motahhare Eslami, Karrie Karahalios, Christian Sandvig, Kristen Vaccaro, Aimee Rickman, Kevin Hamilton, and Alex Kirlik. 2016. First I "like" It, Then I Hide It: Folk Theories of Social Feeds. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2371–2382. https://doi.org/10.1145/2858036.2858494

[21] Motahhare Eslami, Kristen Vaccaro, Min Kyung Lee, Amit Elazari Bar On, Eric Gilbert, and Karrie Karahalios. 2019. User Attitudes towards Algorithmic Opacity and Transparency in Online Reviewing Platforms. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300724

[22] Jennifer Fereday and Eimear Muir-Cochrane. 2006. Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International journal of qualitative methods* 5, 1 (2006), 80–92.

[23] George E Forsythe and Niklaus Wirth. 1965. Automatic grading programs. *Commun. ACM* 8, 5 (1965), 275–278.

[24] Max Fowler, Binglin Chen, Sushmita Azad, Matthew West, and Craig Zilles. 2021. Autograding "Explain in Plain English" questions using NLP. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (Virtual Event) *(SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/3408877.3432539

[25] Batya Friedman and Helen Nissenbaum. 1996. Bias in Computer Systems. *ACM Trans. Inf. Syst.* 14, 3 (July 1996), 330–347. https://doi.org/10.1145/230538.230561

[26] Graham R Gibbs. 2007. Thematic coding and categorizing. In *Analyzing Qualitative Data*. Vol. 2. SAGE, Thousand Oaks, CA, USA, Chapter 4, 38–56.

[27] Douglas Grimes and Mark Warschauer. 2010. Utility in a fallible tool: A multi-site case study of automated writing evaluation. *The Journal of Technology, Learning and Assessment* 8, 6 (2010), 44.

[28] Greg Guest, Arwen Bunce, and Laura Johnson. 2006. How many interviews are enough? An experiment with data saturation and variability. *Field methods* 18, 1 (2006), 59–82.

[29] Shandy Hauk and Angelo Segalla. 2005. Student perceptions of the web-based homework program WeBWorK in moderate enrollment college algebra classes. *Journal of Computers in Mathematics and Science Teaching* 24, 3 (2005), 229–253.

[30] Willett Kempton. 1986. Two theories of home heat control. *Cognitive Science* 10, 1 (1986), 75–90. https://doi.org/10.1207/s15516709cog1001_3

[31] Eileen Kintsch, Dave Steinhart, Gerry Stahl, LSA Research Group LSA Research Group, Cindy Matthews, and Ronald Lamb. 2000. Developing summarization skills through the use of LSA-based feedback. *Interactive learning environments* 8, 2 (2000), 87–109.

[32] René F. Kizilcec. 2016. How Much Information? Effects of Transparency on Trust in an Algorithmic Interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2390–2395. https://doi.org/10.1145/2858036.2858402

[33] Rafal Kocielnik, Saleema Amershi, and Paul N. Bennett. 2019. Will You Accept an Imperfect AI? Exploring Designs for Adjusting End-User Expectations of AI Systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300641

[34] Claudia Leacock and Martin Chodorow. 2003. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities* 37, 4 (2003), 389–405.

[35] John D. Lee and Katrina A. See. 2004. Trust in Automation: Designing for Appropriate Reliance. *Human Factors* 46, 1 (2004), 50–80. https://doi.org/10.1518/hfes.46.1.50_30392

[36] Min Kyung Lee. 2018. Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management. *Big Data & Society* 5, 1 (2018), 2053951718756684. https://doi.org/10.1177/2053951718756684

[37] Bruno Lepri, Nuria Oliver, Emmanuel Letouzé, Alex Pentland, and Patrick Vinck. 2018. Fair, transparent, and accountable algorithmic decision-making processes. *Philosophy & Technology* 31, 4 (2018), 611–627.

[38] Raymond Lister, Elizabeth S Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, Jan Erik Moström, Kate Sanders, Otto Seppälä, Beth Simon, and Lynda Thomas. 2004. A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin* 36, 4 (2004), 119–150.

[39] Karen D Locke. 2000. *Grounded theory in management research*. Sage, Thousand Oaks, CA, USA.

[40] Mike Lopez, Jacqueline Whalley, Phil Robbins, and Raymond Lister. 2008. Relationships between Reading, Tracing and Writing Skills in Introductory Programming. In *Proceedings of the Fourth International Workshop on Computing Education Research* (Sydney, Australia) *(ICER '08)*. Association for Computing Machinery, New York, NY, USA, 101–112. https://doi.org/10.1145/1404520.1404531

[41] Matthew L Meuter, Amy L Ostrom, Mary Jo Bitner, and Robert Roundtree. 2003. The influence of technology anxiety on consumer use and experiences with self-service technologies. *Journal of Business Research* 56, 11 (2003), 899 – 906. https://doi.org/10.1016/S0148-2963(01)00276-4 Strategy in e-marketing.

[42] Laurie Murphy, Renée McCauley, and Sue Fitzgerald. 2012. 'Explain in Plain English' Questions: Implications for Teaching. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (Raleigh, North Carolina, USA) *(SIGCSE '12)*. Association for Computing Machinery, New York, NY, USA, 385–390. https://doi.org/10.1145/2157136.2157249

[43] Corey Palermo and Margareta Maria Thomson. 2018. Teacher implementation of Self-Regulated Strategy Development with an automated writing evaluation system: Effects on the argumentative writing performance of middle school students. *Contemporary Educational Psychology* 54 (2018), 255 – 270. https://doi.org/10.1016/j.cedpsych.2018.07.002

[44] Raja Parasuraman and Victor Riley. 1997. Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors* 39, 2 (1997), 230–253. https://doi.org/10.1518/001872097778543886 arXiv:https://doi.org/10.1518/001872097778543886

[45] Rod D. Roscoe, Joshua Wilson, Adam C. Johnson, and Christopher R. Mayra. 2017. Presentation, expectations, and experience: Sources of student perceptions of automated writing evaluation. *Computers in Human Behavior* 70 (2017), 207 – 221. https://doi.org/10.1016/j.chb.2016.12.076

[46] Amir H Sam, Samantha M Field, Carlos F Collares, Cees PM van der Vleuten, Val J Wass, Colin Melville, Joanne Harris, and Karim Meeran. 2018. Very-short-answer questions: reliability, discrimination and acceptability. *Medical Education* 52, 4 (2018), 447–455.

[47] Mark D. Shermis. 2014. State-of-the-art automated essay scoring: Competition, results, and future directions from a United States demonstration. *Assessing Writing* 20 (2014), 53 – 76. https://doi.org/10.1016/j.asw.2013.04.001

[48] Mark D Shermis and Jill Burstein. 2013. *Handbook of automated essay evaluation: Current applications and new directions*. Routledge, New York, NY.

[49] Raheel Siddiqi. 2013. Impact of automated short-answer marking on students' learning: IndusMarker, a case study. In *2013 5th International Conference on Information and Communication Technologies*. IEEE, IEEE, Karachi, Pakistan, 1–7.

[50] Tovia Smith. 2018. More States Opting to 'Robo-Grade' Student Essays By Computer. https://www.npr.org/2018/06/30/624373367/more-states-opting-to-robo-grade-student-essays-by-computer. Accessed 2020-08-17.

[51] Marie Stevenson and Aek Phakiti. 2014. The effects of computer-generated feedback on the quality of writing. *Assessing Writing* 19 (2014), 51 – 65. https://doi.org/10.1016/j.asw.2013.11.007 Feedback in Writing: Issues and Challenges.

[52] Blase Ur, Pedro Giovanni Leon, Lorrie Faith Cranor, Richard Shay, and Yang Wang. 2012. Smart, Useful, Scary, Creepy: Perceptions of Online Behavioral Advertising. In *Proceedings of the Eighth Symposium on Usable Privacy and Security* (Washington, D.C.) *(SOUPS '12)*. Association for Computing Machinery, New York, NY, USA, Article 4, 15 pages. https://doi.org/10.1145/2335356.2335362

[53] Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education: Research* 2, 1 (2003), 319–330.

[54] Hao-Chuan Wang, Chun-Yen Chang, Tsai-Yen Li, et al. 2005. Automated Scoring for Creative Problem Solving Ability with Ideation-Explanation Modeling.. In *ICCE*. Citeseer, ICCE, Singapore, Singapore, 524–531.

[55] David M Williamson, Xiaoming Xi, and F Jay Breyer. 2012. A framework for evaluation and use of automated scoring. *Educational measurement: issues and practice* 31, 1 (2012), 2–13.

[56] Joshua Wilson and Amanda Czik. 2016. Automated essay evaluation software in English Language Arts classrooms: Effects on teacher feedback, student motivation, and writing quality. *Computers & Education* 100 (2016), 94 – 109. https://doi.org/10.1016/j.compedu.2016.05.004

[57] Allison Woodruff, Sarah E. Fox, Steven Rousso-Schindler, and Jeffrey Warshaw. 2018. A Qualitative Exploration of Perceptions of Algorithmic Fairness. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3174230

[58] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. 2019. Understanding the Effect of Accuracy on Trust in Machine Learning Models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300509