UpGrade: Sourcing Student Open-Ended Solutions to Create Scalable Learning Opportunities

Xu Wang, Srinivasa Teja Talluri, Carolyn Rose, Kenneth Koedinger
Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA
{xuwang, stalluri, cp3a, kk1u}@andrew.cmu.edu

ABSTRACT
In schools and colleges around the world, open-ended homework assignments are commonly used. However, such assignments require substantial instructor effort for grading, and tend not to support opportunities for repeated practice. We propose UpGrade, a novel learnersourcing approach that generates scalable learning opportunities using prior student solutions to open-ended problems. UpGrade creates interactive questions that offer automated and real-time feedback, while enabling repeated practice. In a two-week experiment in a college-level HCI course, students answering UpGrade-created questions instead of traditional open-ended assignments achieved indistinguishable learning outcomes in ~30% less time. Further, no manual grading effort is required. To enhance quality control, UpGrade incorporates a psychometric approach using crowd workers’ answers to automatically prune out low quality questions, resulting in a question bank that exceeds reliability standards for classroom use.

Author Keywords
Crowdsourcing; online education; deliberate practice; open-ended assignment; multiple-choice question.

INTRODUCTION
A key insight that has spawned a new direction in crowdsourcing research called learnersourcing is that learners around the world unwittingly produce content that can be leveraged to create novel learning opportunities. For example, video watching traces [12], video annotations [13, 16, 21], or explanations [22] generated by prior learners were sourced to benefit future learners. In this paper, we explore written homework assignments as a new and powerful input for learnersourcing. Afterall, students are producing great volumes of written content in response to open-ended assignments. We describe how this content can be automatically transformed into online practice activities where student learning is supported through immediate feedback and we present evaluations of the quality of the questions created, the learning outcomes achieved, and time savings for students and instructors. Open-ended assignments are widely used in schools and colleges as formative assessments. They typically involve qualitative feedback offered by instructors, and are designed to inform subsequent learning in contrast with summative assessments, such as exams. At the same time, open-ended assignments require substantive efforts from instructors to grade and provide feedback. Furthermore, the full benefits of this feedback is best realized when it is provided soon after students complete assignments and when they are given the opportunity to incorporate feedback into further practice. However, timely return of detailed feedback is hard to achieve and open-ended assignments are often used as a one-off activity whereby there is little or no chance for deliberate practice on concepts or skills that were not demonstrably mastered.

In this work, we propose UpGrade, a novel learnersourcing approach that delivers scalable and efficient learning opportunities, reducing time commitment from both students and instructors. Following the workflow of UpGrade, instructors can create hundreds of multiple-choice questions from prior student solutions to open-ended problems with minimal effort. UpGrade-created questions also offer real-time feedback for repeated practice. UpGrade can be used as an alternative or primer to traditional open-ended assignments, with more instructional scaffolding towards mastery of the knowledge and skills. UpGrade works by (i) chunking information to be learned into smaller pieces, which allows novices to gradually engage more information; (ii) enabling deliberate practice, which helps novices to develop mastery on knowledge and skills; and (iii) offering immediate and frequent feedback, which helps students stay on track and addresses their errors as they occur.

To evaluate UpGrade in a realistic learning setting, we applied it in a college-level Human-Computer Interaction (HCI) course that teaches user-centered research methods, of which we focused on heuristic evaluation and survey design. In a two-week classroom experiment using a crossover design, we demonstrated that students answering interactive UpGrade-created multiple-choice questions instead of traditional open-ended assignments achieved indistinguishable learning outcomes, while reducing assignment completion time by ~30% and removing the need for instructor grading. This first classroom experiment of UpGrade demonstrates substantial promise for the approach. We also explore crowdsourced methods for evaluating and enhancing the quality of the automatically generated questions. UpGrade incorporates a psychometric method to distinguish reliable versus unreliable question items. Unreliable question items were successfully identified through a validation study with 70 participants on Amazon Mechanical
 Turk. This results in a reliable question bank with an internal consistency that exceeds the standards for classroom use.

In summary, we make the following key contributions:

- New technique: UpGrade, a learnersourcing approach that delivers scalable and efficient learning opportunities, reducing time commitment from both students and instructors.
- Evidence of support for learning: An experiment of UpGrade, demonstrating effective time reduction for students and instructors, while achieving indistinguishable learning outcomes compared to traditional open-ended assignments.
- Approach for quality control: An effective quality control method for automatically selecting high quality learning materials with minimal crowdsourcing effort.

RELATED WORK
Our work extends the frontier of work in an emerging area of crowdsourcing referred to as learnersourcing [10, 12, 13, 16, 21, 22]. The design of UpGrade is motivated by learning theories related to instructional scaffolding [2], worked examples [19], and deliberate practice [5]. To lay a theoretical foundation for our work, in this section we discuss the cognitive processes involved in solving multiple-choice and open-ended problems. From a more practical standpoint we discuss how frequent feedback and deliberate practice are not always affordable for open-ended problems [15]. To address potential concerns that an automated approach to item generation introduces the risk of unreliable or poor quality items, we reviewed established psychometric methods to evaluate test reliability, which informs our quality control approach.

Learnersourcing Techniques
The idea of learnersourcing, proposed and implemented in [10], is a form of crowdsourcing in which learners collectively contribute novel content for future learners while engaging in a meaningful learning experience themselves. For example, LectureScape [12] helps learners navigate online lecture videos using interaction data aggregated over all previous video watchers. ConceptScape [16] generates and presents a concept map for lecture videos through prompting video watchers to externalize reflections on the video. AXIS [22] asks learners to generate, revise and evaluate explanations as they solve a problem, and then presents these explanations to future learners. Other crowdsourcing workflows are designed to extract step-by-step information [13] from how-to-videos or construct subgoals [21] to enhance existing how-to videos.

Prior work used learnersourcing to enhance video watching experience and offer explanations to students. A gap in this literature that our work seeks to fill is that students’ written assignments have not been explored yet as a source for benefiting future learners. Written assignments often take hours of student time to complete, containing rich information, thus could be used a valuable input for learnersourcing.

Worked Examples and Scaffolding
UpGrade addresses two important issues related to design of effective scaffolding, one related to cognitive load and the other related to expert blind spots. First, though open-ended work provides opportunities for authentic learning experiences, a downside is that these rich experiences may consume most of a student’s available cognitive load when they have not mastered the skills and knowledge needed to be successful at the activity [2]. If the problem itself is sufficiently demanding, students may not have enough cognitive resources to learn from solving the problem [17]. Providing instructional scaffolding to a practice activity promotes learning when it helps students practice the target skills at an appropriate level of challenge [3]. Worked examples [19] are one such type of scaffold, which frees up cognitive resources and allows students to see the key features of a problem and analyze the steps and reasons behind problem-solving. UpGrade provides instructional scaffolding in support of open-ended problem-solving through auto-generated worked examples.

A second concern is expert blind spots [18], where the teachers’ expertise makes it difficult for them to anticipate the specific needs of their students. This may prevent instructors from authoring scaffolded learning experiences that take into account all the component skills and knowledge required for complex tasks. On the other hand, prior solutions might provide a complementary source of insight, offering visibility into common mistakes and misconceptions. This motivates the design of UpGrade to decompose student solutions and display the merits or mistakes of the solutions for future students’ reference.

Repeated Practice and Feedback
Deliberate practice, which is focused practice targeting specific skills, assists novices in becoming experts [18]. Research shows that the amount of time a learner spends in deliberate practice rather than more generic practice is what predicts continued learning in a given field [5]. By breaking information down into bite-sized chunks, deliberate practice allows novice learners to gradually engage more information without being overwhelmed [18]. Targeted feedback is also critical during deliberate practice. Many studies have shown that feedback interventions improve learning more than non-feedback ones [14]. Generally, more frequent feedback leads to more efficient learning because it helps students stay on track [8].

However in practice, crafting deliberate practice opportunities with frequent feedback requires careful design and substantive effort from instructors. Furthermore, for open-ended problems, provision of frequent feedback may not be affordable, especially in large-scale classes [15]. In this work, we design UpGrade to offer deliberate practice on open-ended problems without the need for instructors to put in hours of effort in the preparation or during use. One risk of focused, deliberate practice opportunities is that the focused nature might preclude the experience of authentic activities [2]. UpGrade addresses this concern by delivering deliberate practice that is situated within authentic activities.

Quality Control Methods
Prior work has used learner subjective ratings [22] to select high quality content in learnersourcing systems. In this work, we instead explore psychometric methods to evaluate question reliability using student performance data. Common psychometric methods evaluate test reliability by the internal consis-
tency of question items within a test, e.g., using a Rasch model [23], Item Response Theory (IRT) model [7], or Cronbach’s alpha [4]. If question items within a test are consistent in measuring student capabilities or in differentiating knowledgeable and less knowledgeable students, the test is considered reliable and question items are considered to be of high enough quality. On the other hand, if a question item is failing knowledgeable students but favoring less knowledgeable students, the question item is considered to be problematic and needs redesign. Cronbach’s alpha is the most common internal consistency measure, and is incorporated in UpGrade to evaluate the internal consistency of questions generated. An acceptable reliability score (Cronbach’s alpha) for exams is in the range of 0.7-0.95 [20]. As reported in the 2011 TOEFL iBT research report [6], the reliability estimate for TOEFL iBT Speaking and Writing sections are 0.84 and 0.8 respectively, measured by Cronbach’s alpha. We expect a reliability score in the range of 0.7-0.8 to indicate good enough internal consistency of a test and the question items in the test to be reliable.

FORMATIVE STUDY: ASSIGNMENT SURVEY

We first conducted a formative study to understand what commonly-used open-ended assignments look like, and to identify potential cases where sourcing existing examples could be beneficial. We did a content analysis of the assignments of six courses offered to both undergraduate and graduate students in the Human-Computer Interaction (HCI) program at an R1 institution. We used a qualitative approach to examine the learning goals of these assignments and grouped them into several clusters. We identified cases where the skills to be learned in these assignments could be taught through evaluating examples, as shown in Figure 1. We illustrate how we construct the graph below.

The courses we surveyed include two user experience (UX) method courses, two technical (computer science-related) courses, one design course and one learning sciences course. We took a bottom-up approach, mapping out the learning goals and requirements in the assignments. Three clusters of assignments emerged, (i) Solve a problem or generate a solution, which most assignments fall in; (ii) Learn to use a tool, e.g., get familiar with a software, set up a mobile data collection module; and (iii) Share reading reflections and opinions. (ii) and (iii) were less frequent in the sample and were often not graded, here we focus on the main cluster (i).

Problem-solving assignments include both group projects and individual projects. Individual projects are usually intended for skill building, whereas group projects are for practicing skill integration and content generation. For group projects that involve skill building modules, they assemble that of individual projects. Here we only discuss the branch of individual projects. We saw two types of individual problem-solving projects emerging from the data, open-ended problem solving, asking students to generate a solution to a given problem; and doubly open-ended problem solving, asking students to first define a problem and then generate a solution. We highlight the distinction here because they offer different sources for UpGrade to create multiple-choice questions. Among the problem-solving tasks, some have a single success path or a limited set of success paths, e.g., computing the probability of an event using the Naive Bayes model; computing the mean of a variable in a given dataset. Most problems in our surveyed domains (i.e., UX methods, design, learning sciences) do not have a single success path. This also applies to authentic problem-solving tasks in workplaces.

Traditional computer-based tutors such as Assistsments [9] and example tracing tutors [1] were designed for problems with single or limited success paths. UpGrade mainly targets at problems that do not have a single success path. In such problem-solving tasks, students often need to evaluate the solutions they came up with, rationalize why they made the decisions, and revise their solution based on certain criteria. For some domains, the real challenge in solving a problem is to evaluate the quality of a proposed solution rather than to come up with an initial solution. Shifting the practice focus from generating solutions to evaluating existing solutions could be beneficial for learning such skills. We consider such “evaluation-heavy” problem-solving skills (Figure 1) could be exercised well through multiple-choice tasks that emphasize evaluation. We listed example skills that are considered to be “evaluation-heavy” in our survey in Table 1.

<table>
<thead>
<tr>
<th>Course type</th>
<th>“Evaluation-heavy” skill</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical</td>
<td>Propose new features to a model based on error analysis</td>
</tr>
<tr>
<td>Learning sciences</td>
<td>Perform a theoretical cognitive task analysis</td>
</tr>
<tr>
<td>Design</td>
<td>Ideate concept maps and conceptual models</td>
</tr>
<tr>
<td>UX method</td>
<td>Design a survey</td>
</tr>
<tr>
<td>UX method</td>
<td>Heuristic evaluation (critique an interface and come up with redesigns)</td>
</tr>
</tbody>
</table>

Table 1. Examples of “evaluation-heavy” problem-solving skills.

UPGRADE

In this section, we describe UpGrade’s workflow for creating multiple-choice questions from prior students’ open-ended solutions. An overview of the workflow is shown in Figure 2. We illustrate each step using an example to offer a proof of concept that this technique can be applied in practice. The example course we used is an HCI research methods course that has been offered in the department for 5+ years. We refer to the course as UX101 for the rest of the paper. We focused on two topics of UX101 to create questions, Survey Design and Heuristic Evaluation. Both are “evaluation-heavy”
problem-solving skills as categorized in the formative study. Prior offerings of UX101 used one open-ended assignment per topic to help students learn the method. For Survey Design, students were asked to design a survey; for Heuristic Evaluation, students were asked to write a report documenting heuristic problems found for a given website. Past assignment submissions were assessed based on the rubric items shown in Figure 3.

Solution Logging

UpGrade requires structured data of students’ open-ended solutions, which can be logged in different formats. We collected all student assignment solutions under the topics of Survey Design and Heuristic Evaluation that were submitted in the 2015 offering of UX101, with ~100 written assignment solutions per topic. All files were in PDF format, the majority of which had a length of 10+ pages, which is typical for college-level open-ended assignments. The assignment solutions were graded and offered feedback to by peers and TAs through an online platform Coursemark based on the assignment rubric (Figure 3). Feedback data from Coursemark was scraped in association with rubric item for all the solutions. For courses where students’ open-ended solutions are logged in online forms, the next step for solution segmentation will not be necessary.

Solution Segmentation Based on Assignment Rubric

UpGrade then assigns structures to assignment PDF documents by segmenting the solution based on rubric items. For our collected PDF assignment solutions, UpGrade first converts them to HTML files using the Adobe Acrobat API. UpGrade then employs a Python script to segment the HTML files into sections based on DOM tags and text styles (e.g., <h1>, <h2>, <p>). We found this method to be more effective in this segmentation task than using headings or texts. Different students may use different language to describe each section. However, when they start a new section, the DOM tag or text style is always different from the previous section. Moreover, the segmentation technique also associates in-text images with sections, since image DOM tags (e.g., <img>) are inside <p> tags. Each assignment solution file is reorganized into a .txt file with one section per line.

The Survey Design and Heuristic Evaluation assignments followed templates. For example, in the Heuristic Evaluation assignment, students were asked to identify five heuristic problems in a given website. For each heuristic problem, it will be evaluated based on five rubric items, including Description of the problem, heuristic rule Violation, Explanation of why the rule is violated, justification of the Severity of the problem, and a Remedy plan to fix the problem. For solutions whose segmented results matched the rubric items in the template, the segmented sections were automatically associated with each rubric item. However, for solutions that did not follow the exact template, we had to manually align them. For UX101, past instructor and peer feedback were offered in correspondence with the rubric items. Solution segments and feedback offered to the solution were thus automatically matched. From this step, the solution file is reorganized and saved in a local database, an excerpt of which is shown in Table 2.

This manual checking step is a limitation of UpGrade’s current workflow. Potential ways to mitigate this when applying UpGrade in practice include: (i) logging assignment solutions using online forms where structures are predefined, eliminating the need of post-hoc segmentation and metadata association; (ii) abandoning falsely templated solutions when there is a large pool of existing solutions to source from; and (iii) applying advanced approaches to automatically align with the template to minimize the manual checking effort.

![Figure 2. UpGrade’s workflow.](image)

![Figure 3. Rubric items for open-ended assignments on the topic of Survey Design and Heuristic Evaluation.](image)

![Figure 4. Four components used in UpGrade question schemata.](image)

![Figure 5. Example instantiations of the three UpGrade question schemata: (a) Question-Answer, (b) Question-Answer-Explanation, and (c) Answer-Feedback.](image)
Question Creation

We define four components Question, Answer, Explanation, and Feedback (Figure 4) to form question schemata in UpGrade. Question is a question asked in an open-ended assignment, e.g., what are the goals of this survey. In doubly open-ended assignments, students may self-define a Question. Answer is a past student’s answer to a Question. Typical open-ended assignment solutions are composed of many Question-Answer pairs. In some assignments, students are required to offer Explanation to their answers. For assignments that have been graded, instructor or peer Feedback are also collected. Depending on the available data sources, instructors will (i) select a question schema and (ii) specify which sections should be placed into each component in the schema. Examples are given in Figure 5. With the segmented solutions produced (Table 2) and instructor-specified schemata, UpGrade then creates multiple-choice questions automatically. We introduce three question schemata we have defined and explored.

Question-Answer Schema

This schema defines a question with the components Question and Answer. In the example shown in Figure 5 (a), three solution segments including heuristic problem Description, Rule Violation and Explanation were used as the Question. Remedy of the problem was used as the Answer. The distractors were selected from the pool of Remedy that were written for other problems. The example question shown in Figure 6 displays a heuristic problem, and asks question takers to select a remedy that would fix the problem.

Question-Answer-Explanation Schema

When Explanation is available as a data source, it can be used to offer informative real-time feedback in the created question. This schema defines a question with the components Question, Answer, and Explanation. As shown in Figure 5 (b), the heuristic problem Description was used as the Question, and the Rule Violation was used as the Answer. The distractors were selected from the pool of Rule Violation for other problems. The example question shown in Figure 7 describes an interaction scenario of a website, and asks the question taker to identify which heuristic rule is violated. Since the original author offered an explanation on why the rule was violated, the corresponding Explanation is used as feedback to the question taker.

We present another example instantiation of this schema, when there are multiple iterations over a solution. In the survey design assignment, past students designed survey questions, revised them and explained why they made the revision. With this schema, draft 1 of a survey question was used as the Question, revised version of the survey question was used as the Answer. Figure 7 shows an example question created. Both versions of the survey question are displayed, and it asks question taker which version is better. Since the original author explained why they made the revision, the corresponding Explanation is used as feedback to the question taker.

Answer-Feedback Schema

This schema can be used when past instructor or peer Feedback is collected. Since Feedback points to prior stu-
We built a prototype system with Django to render the written solutions of survey goals and asks question takers to select which feedback would apply to each solution.

In this running example, after the segmentation step, we sat down with the UX101 instructor for about two hours in total to decide which question schemata to use and specify the sections to be used in each schema (the same process as shown in Figure 5). We asked the instructor to pick a schema for each rubric item to make sure UpGrade creates multiple-choice questions that cover the full scope of its open-ended assignment counterpart (Table 3). With the instructor-specified schemata, UpGrade creates multiple-choice questions automatically. For example, for HE-1, the specified schema is Q-A-E, also shown in Figure 5 (a). For every (Description, Rule Violation, Explanation) tuple, a question entry is created by selecting three distractors from the pool of Rule Violation. The questions produced by UpGrade are saved in a .csv file.

We built a prototype system with Django to render the questions. The front end of the prototype system looks similar to the interface as shown in Figure 6-9. With one year of past students’ solution, UpGrade created large quantities of multiple-choice questions. To counterbalance, each quiz item had two formats: an open-ended format, and a matched multiple-choice format. For example, a quiz item asked students to identify the design issue of a survey question. The multiple-choice form of the quiz item gave four options for students to choose from (e.g., “leading question”, “asking about averages”), and the open-ended form gave a blank for students to fill.

<table>
<thead>
<tr>
<th>Rubric</th>
<th>Schema</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV-1</td>
<td>A-F</td>
<td>Match instructor feedback to student writing of survey population</td>
</tr>
<tr>
<td>SV-2</td>
<td>A-F</td>
<td>Match instructor feedback to student writing of survey goals</td>
</tr>
<tr>
<td>SV-3</td>
<td>A-F</td>
<td>Match instructor feedback (issues/suggestions) to each survey question</td>
</tr>
<tr>
<td>SV-4</td>
<td>Q-A-E</td>
<td>Compare original and revised question (with UpGrade feedback: student explanation on why they made the revision)</td>
</tr>
<tr>
<td>SV-5</td>
<td>A-F</td>
<td>Match instructor feedback to student design of survey structure</td>
</tr>
<tr>
<td>HE-1</td>
<td>Q-A-E</td>
<td>Decide which heuristic rule is violated in the problem (with UpGrade feedback: student explanation on why it violates the rule)</td>
</tr>
<tr>
<td>HE-2</td>
<td>Q-A</td>
<td>Match severity rating to a student-constructed heuristic problem</td>
</tr>
<tr>
<td>HE-3</td>
<td>Q-A</td>
<td>Match potential remedies to a student-constructed heuristic problem</td>
</tr>
<tr>
<td>HE-4</td>
<td>Q-A</td>
<td>Match potential tradeoffs to a student-constructed heuristic problem and its remedy</td>
</tr>
</tbody>
</table>

Table 3. Course instructor specified a question creation schema for each rubric item in the assignment.

students’ misconceptions and common errors which may repeatedly happen with a new group of students, they can be a good source for creating questions. This schema defines a question with the components Answer and Feedback. As shown in Figure 5 (c), past students’ solution of Survey goals was used as the Answer, and the feedback offered to this solution was used as Feedback. Distractors were selected from the pool of Feedback that have been offered to other solutions. The example question shown in Figure 9 displays past students’ written solutions of survey goals and asks question takers to select which feedback would apply to each solution.

<table>
<thead>
<tr>
<th>Rubric</th>
<th>Trial</th>
<th>Pool</th>
<th>Space</th>
<th>Rubric</th>
<th>Trial</th>
<th>Pool</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV-1</td>
<td>3</td>
<td>18</td>
<td>96</td>
<td>HE-1</td>
<td>30</td>
<td>70</td>
<td>478</td>
</tr>
<tr>
<td>SV-2</td>
<td>3</td>
<td>9</td>
<td>96</td>
<td>HE-2</td>
<td>10</td>
<td>70</td>
<td>478</td>
</tr>
<tr>
<td>SV-3</td>
<td>30</td>
<td>40</td>
<td>NA</td>
<td>HE-3</td>
<td>15</td>
<td>70</td>
<td>478</td>
</tr>
<tr>
<td>SV-4</td>
<td>10</td>
<td>11</td>
<td>NA</td>
<td>HE-4</td>
<td>5</td>
<td>27</td>
<td>91</td>
</tr>
<tr>
<td>SV-5</td>
<td>4</td>
<td>8</td>
<td>96</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Space: number of questions created in total; Pool: number of questions used in the experiment; Trial: number of questions presented to students in each trial.

CLASSROOM EXPERIMENT OF UPGRADE

We conducted a two-week experiment in a college-level HCI course to evaluate UpGrade in comparison with traditional open-ended assignments.

Crossover Experiment Design

We conducted this study in the Spring 2018 offering of the UX101 course, with 28 students enrolled. The course covered one topic (i.e., research method) per week. Instructional activities on each topic included required readings, a 1.5-hour lecture, an open-ended assignment, and a 1.5-hour section. We divided students into two groups, Group A and Group B. Both groups of students did the same regular learning activities (readings, lectures, sections). The only difference was the type of assignment they did. For the topic of Survey Design, Group A worked on the traditional open-ended assignment, and Group B worked on UpGrade-created assignment. Similarly for the topic of Heuristic Evaluation, Group A worked on the UpGrade-created assignment, and Group B worked on the traditional open-ended assignment. Students were given about 7-10 days to finish each assignment.

For students working on UpGrade-created assignments, they logged in to a web-based system with their school ID and completed the assignment online. Student grades on this assignment were determined by how many questions they got right. In the system, students could navigate to different modules to work on the questions in that module. Modules align with the rubric items of the open-ended assignment (Figure 3). For each module, UpGrade produced a large question space. We ranked past student solutions by grade and selected high quality ones to be used in the experiment. The column of Pool in Table 4 indicates the number of questions used in the experiment on each module. Students had unlimited number of attempts at each module, allowing them to work repeatedly on the modules until they achieved a satisfying score. For each trial of a module, Trial number of questions were selected from the Pool (Table 4), giving students different learning opportunities in each trial.

Learning Outcome Measure

We administered a quiz on each topic in class as the learning outcome measure after each assignment was due. The quiz contained 8-12 questions, including both multiple-choice and open-ended questions. To counterbalance, each quiz item had two formats: an open-ended format, and a matched multiple-choice format. For example, a quiz item asked students to identify the design issue of a survey question. The multiple-choice form of the quiz item gave four options for students to choose from (e.g., “leading question”, “asking about averages”), and the open-ended form gave a blank for students to fill.
fill in. In another example, the quiz item asked students to revise a survey question. The multiple-choice form of the quiz item gave four candidate questions for students to choose from, and the open-ended form asked students to revise the question in a text box. By varying the format for each quiz item, two variations of the quiz were created. Both variations had half multiple-choice and half open-ended questions. Students were randomly assigned to one of the variations.

**EXPERIMENT RESULTS**

Learning outcomes were analyzed in a **Condition** (UpGrade-created Multiple-choice vs. Traditional Open-ended) by **Content** (Heuristic Evaluation vs. Survey Design) repeated measures ANOVA. Results indicated a significant main effect of **Content** \(F(1, 26) = 5.76, p = 0.02\), with no main effect of **Condition** \(F(1, 26) = 1.02, p = 0.32\) and no interaction effect. This suggests that students who did UpGrade-created assignment achieved equal learning outcomes in comparison with students who completed traditional open-ended assignments. Surprisingly, we see a trend suggesting that students from UpGrade condition may actually have performed better on the quiz than the Traditional condition, as shown in Figure 10.

In the in-class quiz, students were also asked to self-report the time they spent working on the assignments. We performed another repeated measures ANOVA analyzing assignment completion time by **Condition** and **Content**. Results indicated a significant main effect of **Condition** \(F(1, 24) = 6.55, p = 0.017\), with no main effect for **Content** \(F(1,24) = 0.001, p = 0.97\), and no interaction effect. The average assignment completion time by **Condition** and **Content** is displayed in Figure 11.

Overall, when students did the UpGrade-created assignment composed of multiple-choice questions instead of the traditional open-ended assignment, there was a 28% reduction in assignment completion time, from an average of 6.34 hours \(SD = 3.03\) to 4.56 hours \(SD = 2.63\). The significant results show that this time reduction is substantial. Despite spending less time, students achieved equal learning outcomes. Moreover, the trend in learning outcome even favor the UpGrade condition (Figure 10). Further, UpGrade removed the need of manual grading effort from instructors and TAs.

**User Experience and Feedback**

To better understand user experience and get user feedback to improve UpGrade, we conducted a subsequent interview with the instructor and an in-class interview with the participating students. The instructor liked this approach in that students’ grades were all computed automatically, saving substantial efforts of grading and offering feedback. The instructor further expressed concerns that many students did not do well in the open-ended assignment. “Students are asked to design a survey when they didn’t actually know how to design a survey. Many assignments turned in were in very bad shape and I had to tell the students to go back and redo the assignment.” Additionally, the instructor envisioned future practice where students got to practice with UpGrade first to learn the skills before they went off to generate new content.

Students gave feedback freely during an in-class group interview at the end of a lecture session. Participating students brought up usability issues of UpGrade and suggested ideas for improving the questions in the future. One student commented on the UpGrade heuristic evaluation assignment: “It’s hard to understand the interaction scenario captured by the previous student from a static screenshot. Sometimes we have to guess the intention of the original author.” Another student suggested “In the automated feedback, it gives a more detailed description of the scenario. It’ll be helpful to move some of those texts up to the question stem to illustrate the screenshot.”

The classroom experiment demonstrated UpGrade’s success in saving instructors’ grading time and reducing students’ time to complete a required assignment without sacrificing learning. Subsequent interviews with instructor and students suggested ways to enhance question quality. Though concerns that are inherent to the learnersourcing input (e.g., image quality, text formats) requires more substantial effort to improve, which we will discuss in future work, there is a huge potential to select high quality items taking advantage of the large question pool.

**QUALITY CONTROL**

In this section, we investigate how crowdsourced data can be used to detect reliable versus unreliable question items. More specifically, we ask two research questions, (i) Can we use crowdsourcing to determine which items are unreliable? (ii) If so, how large a crowd is needed and how do we ensure the consistency of crowd workers?
Cronbach’s Alpha to Evaluate Consistency

Cronbach’s alpha [4] is a common psychometric measure of internal consistency across question items within a test. For a test, zero means no consistency at all whereas one indicates perfect consistency. We use it to (i) evaluate the reliability of a set of UpGrade-created questions, and (ii) identify reliable and unreliable questions. To identify reliable and unreliable items, we first compute an overall Cronbach’s alpha on a set of N questions. Then for each of the N questions, if Cronbach’s alpha increases when the item is dropped, the question is indicated to be inconsistent with the rest of the questions, thus being an unreliable item, and vice versa.

MTurk Study

We conducted a validation study on Amazon Mechanical Turk to evaluate the quality of UpGrade-created question items. We focused the validation study on rubric item one in the heuristic evaluation assignment – identify heuristic problems from given scenarios. Figure 7 shows an example UpGrade-generated question on this rubric item. As shown in Table 4, 478 multiple-choice questions were created. We randomly selected 30 questions from the pool to evaluate their quality.

Participants and Procedure

We recruited participants from MTurk located in the US, with greater than 95% assignment approval rate, and more than 500 HITs accepted. Participants first spent 10 minutes reading about heuristic evaluation. Participants then proceeded to complete 30 multiple-choice questions about heuristic evaluation shown in a random order. The task took roughly half an hour to complete, resulting in an hourly pay of ~$8/hour. A total of 70 participants completed the task. On average, participants spent 21 minutes answering all 30 questions, with an accuracy of 50%. To check whether crowd workers were randomly picking responses, we computed a user-user correlation matrix. Results show that among the $70 \times 69 = 2,415$ participant pairs, all pairs had a correlation above 0.85, and 2,405 (99.6%) pairs had a correlation greater than 0.9. This suggests that despite the low accuracy, participants were answering the questions carefully.

Prune Out Unreliable Question Items

The average Cronbach’s alpha for the set of 30 items on the 70 participants dataset was 0.565. The correlation of each item with the total score, and the Cronbach’s alpha after dropping this item are shown in Table 5. Using Cronbach’s alpha as a criterion, 11 items were identified as unreliable items. Removing them resulted in a question bank of 19 items with a Cronbach’s alpha of 0.74, suggesting high internal consistency in assessing student’s heuristic problem identification. The 19 items were thus classified as reliable items.

Question Face Validity Inspection

We further performed a face value inspection analysis to understand what features resulted in unreliable question items. We summarized three reasons when a question is unreliable: (i) Multiple answers could be correct; (ii) There was a lack of description about the scenario, so students had to guess the original content creator’s intention. This was consistent with our interview with students after the classroom experiment;

<table>
<thead>
<tr>
<th>item</th>
<th>corr</th>
<th>alpha</th>
<th>item</th>
<th>corr</th>
<th>alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.53</td>
<td>0.52</td>
<td>16</td>
<td>0.31</td>
<td>0.55</td>
</tr>
<tr>
<td>2</td>
<td>0.51</td>
<td>0.52</td>
<td>17</td>
<td>0.27</td>
<td>0.56</td>
</tr>
<tr>
<td>3</td>
<td>0.49</td>
<td>0.53</td>
<td>18</td>
<td>0.25</td>
<td>0.56</td>
</tr>
<tr>
<td>4</td>
<td>0.45</td>
<td>0.53</td>
<td>19</td>
<td>0.24</td>
<td>0.56</td>
</tr>
<tr>
<td>5</td>
<td>0.45</td>
<td>0.53</td>
<td>20</td>
<td>0.22</td>
<td>0.57</td>
</tr>
<tr>
<td>6</td>
<td>0.44</td>
<td>0.53</td>
<td>21</td>
<td>0.21</td>
<td>0.57</td>
</tr>
<tr>
<td>7</td>
<td>0.43</td>
<td>0.54</td>
<td>22</td>
<td>0.17</td>
<td>0.57</td>
</tr>
<tr>
<td>8</td>
<td>0.42</td>
<td>0.54</td>
<td>23</td>
<td>0.14</td>
<td>0.58</td>
</tr>
<tr>
<td>9</td>
<td>0.39</td>
<td>0.54</td>
<td>24</td>
<td>0.10</td>
<td>0.58</td>
</tr>
<tr>
<td>10</td>
<td>0.39</td>
<td>0.54</td>
<td>25</td>
<td>0.05</td>
<td>0.58</td>
</tr>
<tr>
<td>11</td>
<td>0.38</td>
<td>0.54</td>
<td>26</td>
<td>0.00</td>
<td>0.58</td>
</tr>
<tr>
<td>12</td>
<td>0.38</td>
<td>0.54</td>
<td>27</td>
<td>-0.03</td>
<td>0.57</td>
</tr>
<tr>
<td>13</td>
<td>0.36</td>
<td>0.55</td>
<td>28</td>
<td>-0.04</td>
<td>0.58</td>
</tr>
<tr>
<td>14</td>
<td>0.34</td>
<td>0.55</td>
<td>29</td>
<td>-0.08</td>
<td>0.60</td>
</tr>
<tr>
<td>15</td>
<td>0.32</td>
<td>0.55</td>
<td>30</td>
<td>-0.19</td>
<td>0.62</td>
</tr>
</tbody>
</table>

Table 5. The Pearson’s correlation of each question item with the total score and the average Cronbach’s alpha for the set when the item is dropped. Higher correlation and lower Cronbach’s alpha indicates higher reliability.

Cost-effectiveness of Quality Control

With 70 crowd workers’ performance data, we successfully identified 11 unreliable items in the 30-question sample. However, it may be unrealistic to recruit a large population of crowd workers to prune out unreliable question items for classroom use. With the collected MTurk dataset, we further investigated the minimal crowd size requirement for cost-effective quality control. We used the identified 19 reliable and 11 unreliable items as an approximation of ground truth. We then conducted experiments with varying crowd sizes from 5 to 70. We computed the accuracy for each experiment against the ground truth using the formula: (True Positives+True Negatives) / Number of Items. For each crowd size, we did 100 iterations of random sampling, and computed the average accuracy. The change of accuracy by crowd size is displayed in Figure 12. We can already do a decent job of differentiating reliable vs. unreliable items with a crowd size of 25 (accuracy = 0.8), and with a crowd size of 50, accuracy can reach 0.9.

Further, we investigated the crowd size requirement if the goal was to identify a subset of unreliable items. We ranked all 30 items that have been tested by their score correlation with the total score (Table 5), and used this as an approximation of the question quality ranking’s ground truth. We then conducted experiments to investigate the crowd size requirement.
for identifying the X least reliable items in our sample. In the experiments we varied two variables, the crowd size, and the X least reliable items in the sample. For each combination of crowd size and X, we did 100 iterations of random sampling and computed the average accuracy on detecting the X least reliable items. Figure 13 shows the average accuracy for each experiment. When the goal was to detect the one least reliable item, we achieved an accuracy of 0.95 with only five students. When the goal was to detect the three least reliable items, we achieved an accuracy of 0.8 with five students. These experiments demonstrated that more cost-effective quality control can be achieved depending on the needs.

Summary
Through quality control, we successfully identified 19 reliable multiple-choice questions that are highly consistent, with an average Cronbach’s alpha of 0.74. Considering the recommended reliability scores for exam use is 0.7-0.95 [20], the resulted question bank meets the criteria for classroom use. Proportionally, with the existing assignment data we have for UX101, we estimate UpGrade can output ~300 reliable multiple-choice questions on one rubric item after quality control. From a time consumption standpoint, if we hire crowd workers for quality control, assuming we prune out six questions in each 30-question set with 10 workers, UpGrade can generate 100 reliable questions with a minimal of 13 hours (10 × 4 × 20 minutes) of crowd workers’ time.

In contrast, it would take far more than 13 hours for instructors to write 100 reliable multiple-choice questions with feedback. Consider that the average time students spent to complete the open-ended assignment is 6.3 hours (as in the classroom experiment), which only includes five heuristic problems and corresponds to 5 multiple-choice questions. From a cost standpoint, it is nevertheless to mention it requires far more than 13 × 8 = $104 to hire an expert to generate 100 practice questions. On the other hand, it might not be necessary to hire crowd workers for quality control. As more students use UpGrade, student performance data can be incorporated to prune out unreliable items, though with the risk of presenting low quality materials to students.

DISCUSSION, LIMITATIONS AND FUTURE WORK
In this section, we discuss the limitations and potential future directions to enhance the question quality and the learning benefit of UpGrade.
ACKNOWLEDGEMENT
This work was funded in part by NSF grant ACI-1443068. In addition, we thank Ellen Ayoob, Jim Morris, Juho Kim, Paulo Carvalho, Steven Dang, Anhong Guo and all participants.

REFERENCES


