# Gradescope: a Fast, Flexible, and Fair System for Scalable Assessment of Handwritten Work

**Arjun Singh**
Gradescope, Inc.
Berkeley, CA
arjun@gradescope.com

**Sergey Karayev**
Gradescope, Inc.
Berkeley, CA
sergey@gradescope.com

**Kevin Gutowski**
Gradescope, Inc.
Berkeley, CA
kevin@gradescope.com

**Pieter Abbeel**
UC Berkeley
Berkeley, CA
pabbeel@cs.berkeley.edu

## ABSTRACT

We present a system for online assessment of handwritten homework assignments and exams. First, either instructors or students scan and upload handwritten work. Instructors then grade the work and distribute the results using a web-based platform. Our system optimizes for three key dimensions: speed, consistency, and flexibility. The primary innovation enabling improvements in all three dimensions is a dynamically evolving rubric for each question on an assessment. We also describe how the system minimizes the overhead incurred in the digitization process. Our system has been in use for four years, with instructors at 200 institutions having graded over 10 million pages of student work. We present results as user-reported data and feedback regarding time saved grading, enjoyment, and student experience. Two-thirds of responders report saving 30% or more time relative to their traditional workflow. We also find that the time spent grading an individual response to a question rapidly decays with the number of responses to that question that the grader has already graded.

### Author Keywords

education; learning assessment; rubric-based grading; computer-assisted instruction; scaling large courses

## INTRODUCTION

Over the past few years, course sizes have gone up significantly at many higher education institutions. Although there have been many recent innovations in teaching that aim to help scale up courses (e.g. MOOCs), there are two primary components involved in teaching that are difficult to scale.



*L@S 2017,* April 20 - 21, 2017, Cambridge, MA, USA



ISBN 978-1-4503-4450-0/17/04…$15.00

DOI: http://dx.doi.org/10.1145/3051457.3051466

The first is personal interaction with instructors and tutors. The second is fair, informative assessment of student work without compromising on question quality. This is our focus.

Assessing student work is one of the most tedious and time consuming aspects of teaching. It is also one of the most important, being a primary feedback mechanism for students. One solution to scaling course sizes is to simply give assessments that can be graded automatically, such as multiple choice exams. Although these assessments can be valuable, there are many concepts that are better assessed by free response questions than multiple choice questions. Our system allows instructors to use exactly the same questions in a 1000 student course that they would in a 25 student course, and grade them quickly and consistently.

The primary benefits of our system are:

1. **Speed:** most users report that their grading is sped up by a third, versus paper-based grading.
2. **Consistency:** most users report they are able to grade more fairly while helping students learn from mistakes and providing transparency in grading.
3. **Flexibility:** users can modify rubrics as they encounter new mistakes, or revise earlier evaluations.

Our system is publicly available[1] and has been used in over two thousand higher-ed courses.

In this paper, we first give an overview of related work in Section 2. We then describe the system in detail in Section 3, and provide results for how the system performs in Section 4. Lastly, we discuss future work and share concluding thoughts in Section 5 and Section 6.

## RELATED WORK

Grading assignments has always been a major pain point and bottleneck of instruction, especially in large courses and in distance education. The challenges of scaling grading are twofold. The more students there are, the more graders are needed

[1] https://gradescope.com

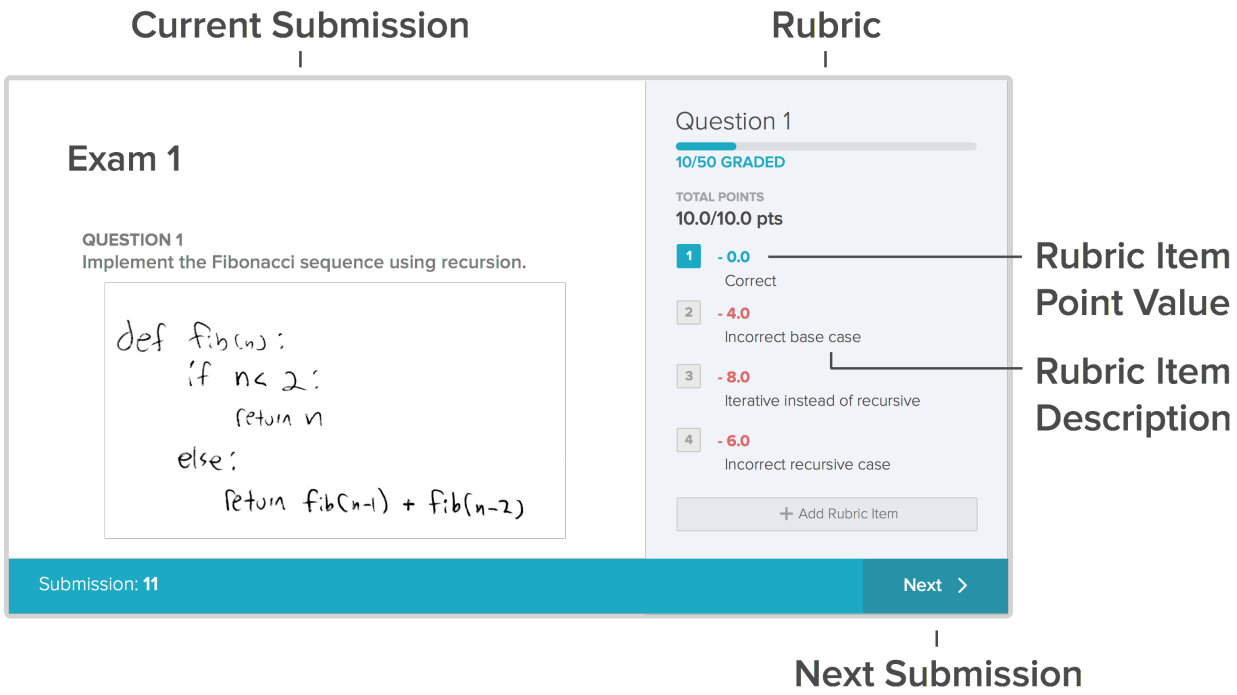


Figure 1: Our system's grading interface, simplified slightly and annotated for publication. On the left, the grader sees a single student's submission to the question they're grading. On the right, they see the rubric, composed of multiple rubric items that each have point values and descriptions. When finished grading, graders navigate to the next submission for the same question.

to deliver feedback in a timely manner. But the more graders are involved, the less consistent grading tends to become.

Automating assessment is one answer to the problem. There is a body of work aiming to automate more parts of computer science, engineering, and writing assessment. For computer science courses, focus has been on autograding of programming projects [11, 8, 3] and automated plagiarism detection [14]. For general engineering courses, notable new efforts include automatic grading of engineering drawings [10]. For grading essays and other student writing, software such as Canvas Speedgrader exists for entirely manual scoring, and both research and commercial systems for autograding have existed for over a decade [18, 4].

However, there are only a few systems focused on grading paper-based work. The one most similar to our system is a tablet-based paper exam grading (T-Pegs) system described by Bloomfield and Groves in 2008 [2]. Bloomfield described improvements to this system in 2010 [1]. In the original system, graders simply assigned a point value to each page in the scan. In the followup, the system was extended to allow graders to give a point value for each question, along with some textual feedback.

Park and Hagen [12] describe a fax-based system for managing large quantities of work being graded by hand on paper in a distance education setting. Schneider [15] describes a system for grading handwritten homework, in which students upload scans of their work to be graded. Instructors of a large business course also report their initial experiments with online marking of scanned assesments [5].

Our system has much in common with these: student work is scanned in and then digitally assessed, and graders can be in any physical location, and can grade in parallel. However, we have several key differentiators from work listed above:

1. We allow a much richer form of feedback due to the rubric. Rather than giving a single score with a bit of text as feedback, students are graded on a rubric, enabling transparency and consistency. Rubric-based grading, including sharing of the rubric with students, has been shown to both increase inter-grader reliability and improve student educational outcomes [17, 16, 13, 9].

2. We do not require the exams to be preprocessed in any way. An instructor can grade exams with our system without modifying their existing exams. Systems described in [1, 5] require exams to have special frontmatter sections in order to match scans with students.

3. We support both exams and homework in a single system.

4. We allow students to securely view their work online, potentially as soon as grading is completed. Prompt delivery of informative feedback on the student's work has been shown to increase learning in students [7].

5. Instructors can choose to allow students to submit regrade requests directly to their graders, to cut down on time spent during office hours on such requests.

6. We automatically expose detailed statistics to instructors, including which mistakes were made most frequently on every question on an assessment.

## SYSTEM DETAILS

In this section, we describe our system in detail. First, we explain how to set up an assessment for grading. Next, we discuss how we minimize the overhead associated with scanning and digitizing students' paper assignments. We then describe the grading process, including the dynamic rubrics. We explain how students can view their graded work and request clarifications. Lastly, we describe how the system enables instructors to analyze their students' performance.

### Setup

Assignments are generally one of two types: (1) worksheet-style, fixed-length assignments in which every student writes on a template and submits the same number of pages, and (2) variable-length assignments, in which students might be asked to answer questions out of a textbook and use an arbitrary number of pages.

Our system supports both types of assignments effectively. However, for clarity of explanation, we will assume that the instructor has a fixed-length assignment. We will describe the process for an exam (as opposed to a homework assignment), such that it is the instructors responsibility to digitize the students' work.

In order to optimize the workflow for a worksheet-style exam, the instructor first uploads a *template* of their exam, and then sets up the *assignment outline*. The template is simply a blank PDF of the exam. The assignment outline consists of the list of questions on the assignment, their point values, and the region on the template that corresponds to each question. They select the regions by drawing boxes on the exam, as shown in Figure 2. They also draw a box around the region where students write their name on the exam, which allows the instructor to quickly label each exam with a student, as described in Section 3.2.2.

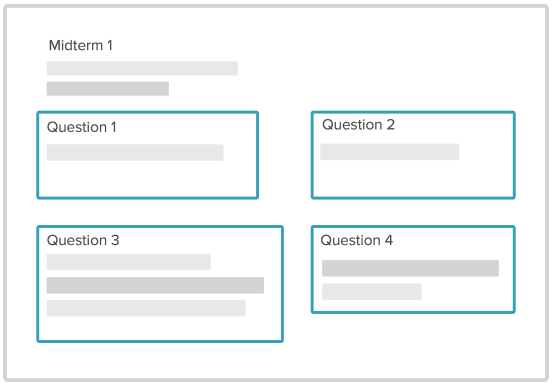


Figure 2: When creating the assignment outline, the user draws boxes corresponding to where on the page the student writes their responses to each question.

### Scanning

One important constraint that we built into our system is that exams did not need to be altered in order to be graded online. Other systems, such as described in [1], require the exams to be preprocessed with bubble sections, QR codes, or other markers. For our system, the pre-exam workflow is exactly identical to grading on paper.

After the students write their answers on the exam template and return their work to the instructor, the exams have to be scanned and associated with the students. This is the largest upfront cost of our system. We reduce the time spent in this step of digitizing the exams in two ways.

#### *Scanning in Batches*

First, we allow instructors to scan the exams in batches. This can dramatically reduce the amount of time a scanner is sitting idle, waiting to be fed with the next exam. The system then automatically suggests how to split the batch into individual exams, as shown in Figure 3. The user is able to confirm that the split is correct, or merge, rearrange, or reorder pages.

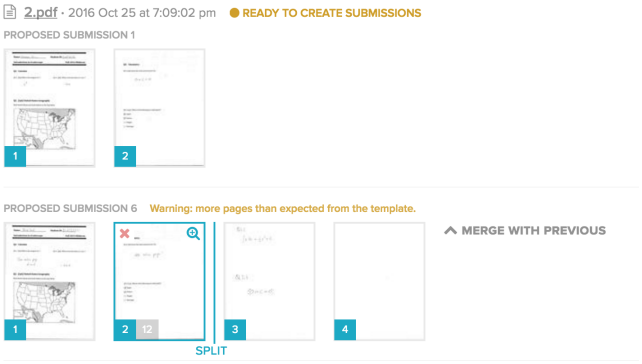


Figure 3: Screenshot of the scan splitting interface. The user is able to confirm that the split is correct, or alter the proposed split.

#### *Assigning Names*

We make the name assignment step fast, without automating it. This leads to a far lower error rate in name assignment than that shown by some of the automated systems [1]. In order to minimize time spent on this step, we show only the part of the page with the student's name on it, and autocomplete based on the roster, as shown in Figure 4.


Figure 4: Screenshot of the submission naming interface. We show the area of the scan where the students write their names, and then autocomplete what the user has typed against the course roster.

### Grading With a Rubric

Once the exam is set up and the scans are uploaded and split into submissions, users can start grading. Naming submissions is not required prior to grading. Graders can grade different responses to the same question (the system ensures that graders

do not evaluate the same student) – or they can grade different questions altogether. They are able to grade in parallel, and they do not need to be in the same physical location.

The grading interface is shown in Figure 1. It consists of a single student's answer to a single question, as well as a *rubric* that is built up as the instructor grades.

The rubric is composed of one or more *rubric items*. Each rubric item has a point value and a description associated with it, as illustrated in Figure 5. The rubric can be subtractive (rubric items correspond to point deductions, or mistakes), or additive (rubric items correspond to point additions), and it can have a point value floor and ceiling. For clarity, we assume that the rubric is subtractive in this paper.

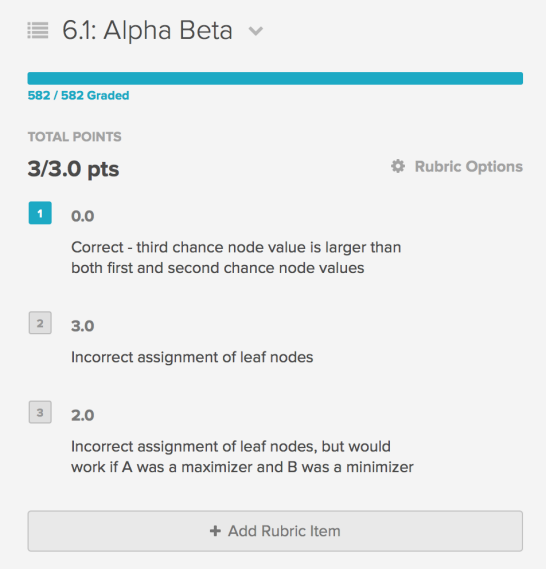


Figure 5: Screenshot of a rubric. The first item in the rubric is applied to this particular student, and the student received full credit.

One of the most common grading workflows is detailed below:

1. Look at the student answer and find any mistakes made.
2. If there are new types of mistakes that aren't yet in the rubric, create rubric items for each new type of mistake.
3. Apply each rubric item corresponding to each mistake made by the student.
4. Go to the next ungraded student answer *for the same question*.
5. Repeat steps 1-4 until this question is graded, and then move to the next question.

One important attribute of the system is that the rubric is *dynamic*. As graders find new types of mistakes, they can add new rubric items to the rubric. Furthermore, if a grader realizes that the point value associated with a rubric item should be changed, they can do so, and our system will retroactively update the grades given to all previous students for that question according to the updated rubric. This process is illustrated in Figure 6.

Although graders can build rubrics and grade work in any order, the above workflow has a few key benefits. First, it allows graders to focus on a single question at a time. Rather than needing to understand the rubric for all questions on the exam, they only need to worry about the question they are grading. Second, it helps enforce consistency: if all new mistakes are added to the rubric as they are seen, then graders can be confident that each instance of a mistake had the correct number of points deducted.

#### *Additional grading features*

Although the bulk of grading actions on our platform occur via the rubric, there are two more ways to provide feedback to students while grading: comments and free-form annotation of the scan. The grader is always able to leave a comment on a student submission that is only visible to that student. This feature is useful when, for example, an answer is wrong in a singular way that does not merit inclusion in the rubric. Alongside the comment, the grader can adjust the total score however they would like. Additionally, the grader can annotate the scan area with a free-form pen tool, which is especially useful when grading on a touch-screen or tablet device.

### Analysis of Student Performance

In addition to simply saving a significant amount of time, digital grading enables analysis of student performance that is quite tedious with traditional paper-based grading. We can track per-assignment and per-question statistics, as the Bloomfield system also does [1]. In addition to this basic level of analysis, we additionally enable rubric-level statistics.

Because grading is done using a rubric in our system, instructors are able to see exactly *which* mistakes were made most often by students in our rubric-level statistics view (see Figure 7). This kind of analysis is nearly impossible with traditional paper-based grading, and it can give valuable insight into specific misconceptions that students developed.

### Distribution of Work to Students

Once satisfied with how the exam is graded, the instructor can securely return the work to the students with a single click. In accordance with privacy regulations such as FERPA, students only have access to their own work. Students can see the scan of their exam and their score on every question.

By default, students can see which entries in the rubric applied to their answer, as well as the rest of the rubric. This allows them to understand all of the ways students could earn or lose points on the question. Although this is the default, the instructor can choose to limit student visibility of the rubric to either nothing or only applied items. Furthermore, the instructor is able to see whether each graded submission has been reviewed by the student.

### Handling Regrade Requests

Often, students will have questions or feel that mistakes have been made in grading their work. With paper-based grading,

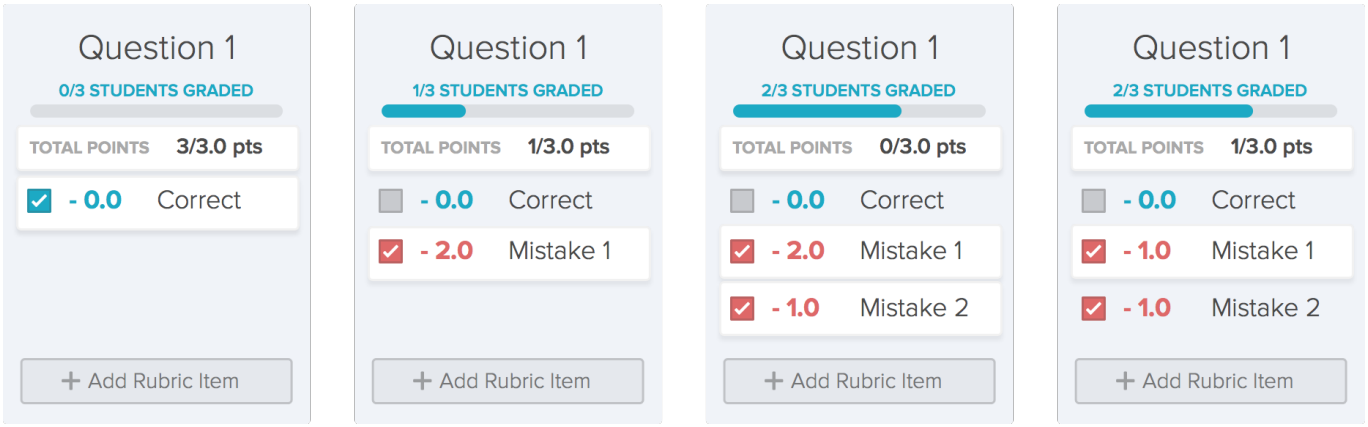


Figure 6: Illustration of the dynamic rubric. At first, the rubric contains only a single item. When a grader finds a new mistake, they add an item to the rubric. More than one rubric item may be applied to the same student's answer. The point value of any rubric item can be adjusted at any time, and the system will retroactively update the score assigned to any previously graded answers.

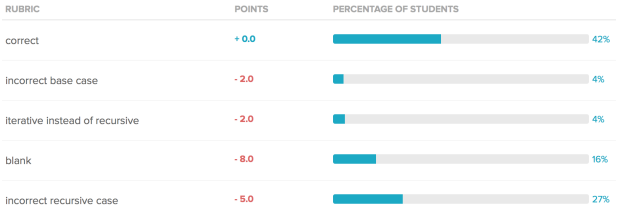


Figure 7: Screenshot of rubric statistics. Each rubric item is shown, along with its point value and the percentage of student answers it was applied to.

this typically leads to an unwieldy process, in which a student will email a request to their instructor and/or show up at their office hours. In large classes, the instructor they interact with is often not the person who graded that student's response. Getting the request into the hands of the correct grader often takes several days, leading to a long turnaround time for the student.

Our system tracks which grader graded each student's answer, and will notify the grader when a student requests a regrade. The status of each request is centrally tracked, so the instructor can confirm whether all outstanding requests have been handled. The student is also notified when a request is handled.

## RESULTS

We analyze our system in two ways: a user survey, and statistics about usage.

### Survey Results

We asked a series of questions to our instructor user base in 2014, consisting of faculty and teaching assistants. We report the detailed survey results in Tables 1 through 7. The majority of users agree or strongly agree that the system helps them grade more fairly, faster, and more enjoyably. Additionally, the majority of users agree or strongly agree that the system simplified the regrade request process and helped students learn from the feedback.

To quantify the time savings of our system, we asked "How much time do you save grading with our system versus grading on paper?" As reported in Table 7, 67% of the users said that they cut down grading time by at least 30%.

| | |
|---|---|
| Strongly Agree | **32 (46.4%)** |
| Agree | 23 (33.3%) |
| Neutral | 12 (17.4%) |
| Disagree | 2 (2.9%) |
| Strongly Disagree | 0 (0%) |

Table 1: Does the system help you grade more fairly?

| | |
|---|---|
| Strongly Agree | **42 (60.9%)** |
| Agree | 18 (26.1%) |
| Neutral | 4 (5.8%) |
| Disagree | 4 (5.8%) |
| Strongly Disagree | 1 (1.4%) |

Table 2: Does the system save you time in grading?

| | |
|---|---|
| Strongly Agree | 18 (26.9%) |
| Agree | **22 (32.8%)** |
| Neutral | 20 (29.9%) |
| Disagree | 4 (6%) |
| Strongly Disagree | 3 (4.5%) |

Table 3: Does the system make grading more enjoyable?

### Data Analysis

Our system has been used at over 200 different schools to grade over 10 million pages of student work. As shown in Figure 8, course sizes on our platform range from typical K-12 sizes of 20-30 students per course to over 1700 students in our largest course.

| | |
|---|---|
| Strongly Agree | **28 (41.2%)** |
| Agree | 22 (32.4%) |
| Neutral | 8 (11.8%) |
| Disagree | 9 (13.2%) |
| Strongly Disagree | 1 (1.5%) |

Table 4: Does the system simplify regrade requests?

| | |
|---|---|
| Strongly Agree | 12 (17.6%) |
| Agree | **31 (45.6%)** |
| Neutral | 19 (27.9%) |
| Disagree | 5 (7.4%) |
| Strongly Disagree | 1 (1.5%) |

Table 5: Does the displayed rubric help your students learn more from their mistakes?

| | |
|---|---|
| Strongly Agree | 26 (38.2%) |
| Agree | **30 (44.1%)** |
| Neutral | 10 (14.7%) |
| Disagree | 2 (2.9%) |
| Strongly Disagree | 0 (0%) |

Table 6: Does the system offer transparency to my students about the grading scheme?

| **Time saved vs. traditional grading** | **% of users** |
|---|---|
| > 10% | 91% |
| > 20% | 88% |
| > 30% | 67% |
| > 40% | 49% |
| > 50% | 40% |
| > 60% | 19% |
| > 70% | 12% |
| > 80% | 7% |
| > 90% | 3% |

Table 7: How much time do you save grading with the system?

Assignments on our platform range from just a question or two to over 30 questions, as shown in Figure 9. Exams and quizzes (instructor-scanned assignments) tend to have more questions than homework (student-scanned assignments). Very few questions on our platform have been of the multi-page essay type.

*Rubric usage statistics*

For this analysis, we look at questions with at least 40 graded submissions.

First, we examine how instructors set up question rubrics: do they use the rubric to mark different types of mistakes, thereby providing feedback, or do they essentially ignore this feature of our system, and simply mark "correct/incorrect"?

Figure 10 shows that most questions average 5.6 rubric items, with standard deviation of 3.9. The median number of rubric items is 5. There are no meaningful differences in these statistics between subtractive and additive rubrics. More questions have 8 or more rubric items than have 2 or less. This shows

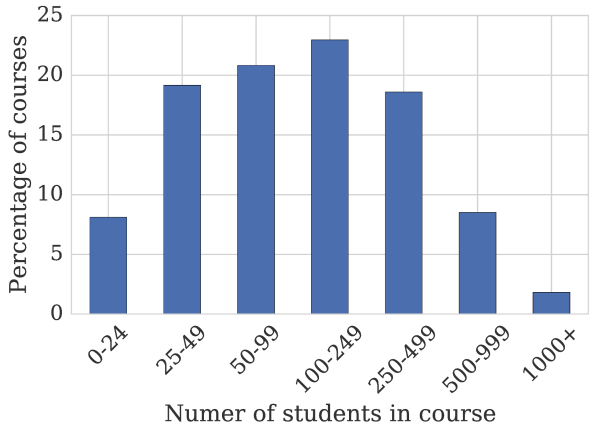


Figure 8: Percentage of courses in our system by number of students in the course.

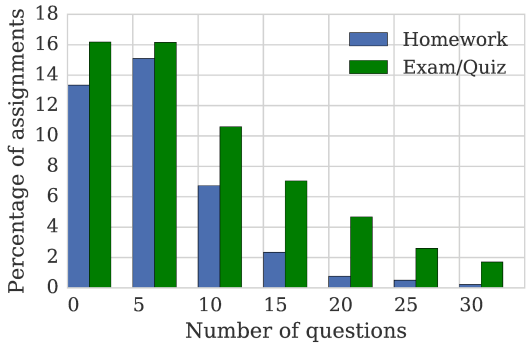


Figure 9: Histogram of the number of questions per assignment.

that instructors choose to give more detailed feedback than simply marking correct or incorrect, and that they do this for both additive and subtractive rubrics.

*Grading time per submission*

Second, we examine the relationship between time spent grading the average student answer and the number of student answers already graded. If the dynamic rubric works as described, then time per answer should go down with more and more answers graded.

For this plot, we look at a random set of 100 courses in Computer Science subjects in which our system was used for at least 2 assignments, each with at least 50 students. There are 596 assignments composed of 7,710 questions assessed in these courses. The vast majority of the questions (6,258) were graded by one person – but some questions were graded by many people (as many as 17 for one question).

Figure 11 shows that time spent per student answer rapidly decays with number of answers graded. Some details for this plot are in order. "Grading time" is measured as the interval between successive grading actions, per grader per question. Intervals of more than ten minutes are filtered out, as they

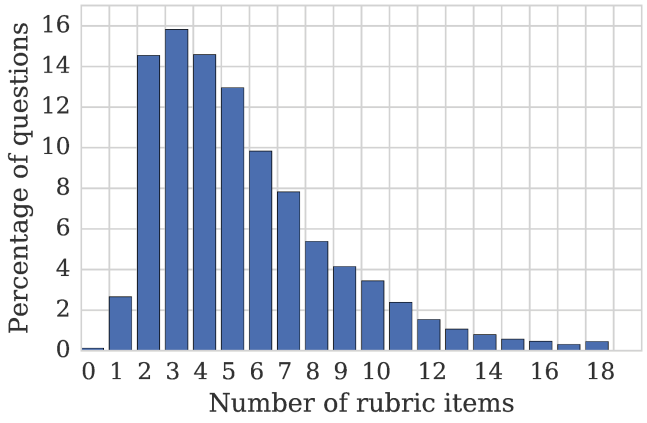


Figure 10: Histogram of the number of rubric items per question.

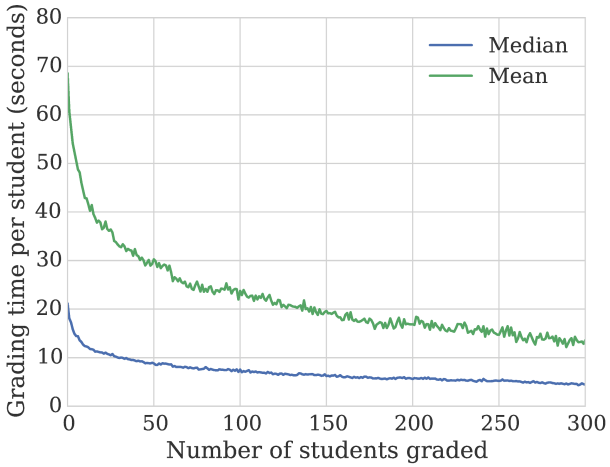


Figure 11: Time spent grading each submission vs number of graded submissions.

correspond to boundaries between distinct grading "sessions." Arranging the grading times in order of their execution, we compute the mean and median across all graders and questions.

*Total grading time*

Total grading time per assignment is a useful metric that is easy to obtain with our system. We plot it relative to the number of student submissions in Figure 12. The median assignment in our sample dataset has 14 questions, 141 student submissions, and took 14.6 person-hours to grade.

## FUTURE WORK

Perhaps the most interesting consequence of grading digitally is that all of the grading data is also digitized, in a form that enables easy analysis. We plan to build tools that enable both students and instructors to benefit from this data.

First, we can allow users to tag questions with concepts. If all questions on the homework assignments and exams in a course are tagged with concepts, we can provide a dashboard illustrating how every student in a course is performing on each concept. In addition to the instructor, who will be able to adjust their teaching accordingly, students could benefit from seeing similar data about their performance. Roughly speaking, there are two ways for a student to receive an 80% on an assignment: get 80% partial credit on all questions, or to entirely miss one question out of five. If the missed question corresponds to one particular concept, the study plan for the next assignment should be very clear.

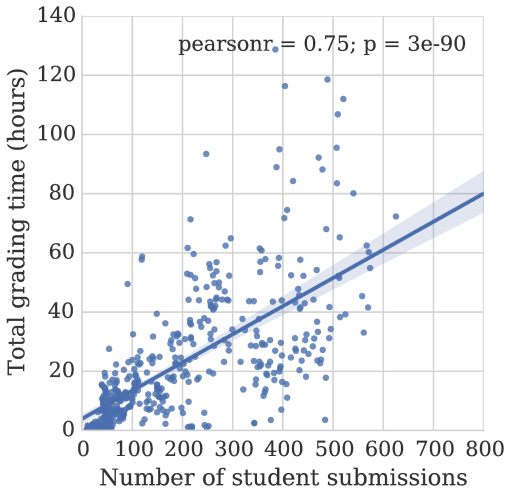


Figure 12: Total time spent grading the assignment in person-hours vs number of student submissions.

We also aim to point out to instructors which of their questions might be misleading. For this, some extensions of item response theory [6] are necessary for our rubric-based data.

Many of the student users of our system are enrolled in several courses using the system. This allows us to longitudinally track student performance throughout their academic careers, and yield insights into curricular development.

Lastly, we would like to support open-response assessment in online courses. In a brief pilot that allowed online students in UC Berkeley's CS188x edX offering to submit the same open-response, paper-based final as in-class students, we learned that MOOC students welcomed the opportunity to be tested more rigoruously than their usual automatic assessment allowed. We look forward to integrating with more MOOCs.

## CONCLUSION

We described an online system for fast, fair, and flexible grading of handwritten assignments. In four years of usage, over 10 million pages of student work have been graded, corresponding to over 100 thousand questions. In survey, instructors report that our system enables them to provide higher quality feedback in less time than with traditional paper-based grading. With no additional effort, instructors also get detailed, actionable statistics on assignment, question, and rubric levels.

## REFERENCES


1. Aaron Bloomfield. 2010. Evolution of a digital paper exam grading system. In *Frontiers in Education Conference (FIE)*. IEEE.
2. Aaron Bloomfield and James F Groves. 2008. A tablet-based paper exam grading system. In *ACM SIGCSE Bulletin*, Vol. 40. ACM, 83–87.
3. Brenda Cheang, Andy Kurnia, Andrew Lim, and Wee-Chong Oon. 2003. On automated grading of programming assignments in an academic institution. *Computers & Education* 41, 2 (2003), 121–131.
4. Ronan Cummins, Meng Zhang, and Ted Briscoe. 2016. Constrained Multi-Task Learning for Automated Essay Scoring. Association for Computational Linguistics.
5. Andrew Eberhard and Donald Sheridan. 2015. The Transition to Online Marking in Large Classes. In *EdMedia: World Conference on Educational Media and Technology*, Vol. 2015. 371–376.
6. Susan E Embretson and Steven P Reise. 2013. *Item response theory*. Psychology Press.
7. John Hattie. 2008. *Visible Learning: A Synthesis of Over 800 Meta-Analyses Relating to Achievement*. Taylor & Francis. `https://books.google.com/books?id=c2GbhdNoQX8C`
8. Michael T Helmick. 2007. Interface-based programming assignments and automatic grading of java programs. In *ACM SIGCSE Bulletin*, Vol. 39. ACM, 63–67.
9. Anders Jonsson and Gunilla Svingby. 2007. The use of scoring rubrics: Reliability, validity and educational consequences. *Educational research review* 2, 2 (2007), 130–144.
10. Youngwook Paul Kwon and Sara McMains. 2015. An Automated Grading/Feedback System for 3-View Engineering Drawings using RANSAC. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*. ACM, 157–166.
11. David J Malan. 2013. CS50 sandbox: secure execution of untrusted code. In *Proceeding of the 44th ACM technical symposium on Computer science education*. ACM, 141–146.
12. James Park and John Hagen Jr. 2005. Managing Large Volumes of Assignments. *EDUCAUSE Quarterly* (2005).
13. Y Malini Reddy and Heidi Andrade. 2010. A review of rubric use in higher education. *Assessment & Evaluation in Higher Education* 35, 4 (2010), 435–448.
14. Saul Schleimer, Daniel S Wilkerson, and Alex Aiken. 2003. Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, 76–85.
15. Susan C Schneider. 2014. "Paperless Grading" of Handwritten Homework: Electronic Process and Assessment. In *ASEE North Midwest Section Conference*.
16. Donald Sheridan and Lesley Gardner. 2012. From Cellulose to Software: The Evolution of a Marking System. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Vol. 2012. 454–461.
17. D.D. Stevens and A. Levi. 2005. *Introduction to Rubrics: An Assessment Tool to Save Grading Time, Convey Effective Feedback, and Promote Student Learning*. Stylus Pub. `https://books.google.com/books?id=LIxWgDn8_N0C`
18. Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education* 2 (2003), 319–330.