# An Integrated Framework for the Grading of Freeform Responses

Piotr F. Mitros*[†][‡][¶] Vikas Paruchuri*[†][§] John Rogosic[†] Diana Huang[†]
**Both authors contributed equally.*
[†]*edX, Cambridge, MA, USA*
[‡]*Department of EECS, Massachusetts Institute of Technology, Cambridge, MA, USA*
[§]*vik@edx.org*
[¶]*pmitros@edx.org*

### Abstract

Massive open online classrooms (MOOCs) have the potential to educate millions of people around the world. Initial MOOC courses were in science and engineering disciplines, where the problems involve constrained choices and can easily be graded automatically. MOOCs must still find ways to deal with essays and short answers, which are required for classes in humanities and the social sciences, and are useful to a variety of other disciplines. Three of the general techniques for evaluating freeform content are self assessment, peer assessment, and AI assessment. We describe how these approaches are implemented in the edX platform, and we present an approach which integrates scoring and feedback from the three techniques in order to maximize accuracy and minimize student and instructor effort. This combined approach has the potential to offer greater accuracy and better feedback with less overhead than any technique in isolation. We present a preliminary implementation of the integrated approach, as built into the edX platform, as well as results from pilot experiments with self-assessment and peer grading.

## I. Introduction

Massive open online classrooms have the potential to give hundreds of millions of people around the world access to the same high-quality education available to residential students at elite institutions while both improving residential education and providing tools that help us understand how students learn.

First generation MOOCs offered a very limited set of assessment types. For instance, the original Stanford AI course[1] was limited to numeric, true/false, and multiple choice answers. This limitation placed substantial constraints on the pedagogy that could be used. Second generation MOOCs began introducing richer assessment tools. For example, 6.002x, the first edX course[2], provided rich tools for automatic grading of complex problems, such as circuit schematics and symbolic equations, while courses like 6.00x and CS188x had rich autograders for computer code. These tools allow for a wide range of design problems and open-ended questions to be offered, but are still primarily limited to STEM disciplines.

As MOOCs move to offer courses in humanities and liberal arts, a range of new assessment techniques are being developed. Many of the more innovative involve substantial changes in course delivery. In this paper, we focus on techniques which lend themselves to assessment of conventional residential open-ended problems. Sections II, III, IV, V describe the isolated techniques, best practices, and how those are embodied in the edX platform. Section VI lays out a general formulation for the problem of integrating those techniques. Section VII describes a simple implementation of an integration, as embodied in the edX platform.

## II. Self-Assessment

In self-assessment, a student is first asked to answer a question, after which they are shown a rubric and asked to assess their own answer. Self-assessment works very well in situations where a problem has a clear rubric, and where students have the requisite knowledge to grade their own work.

Students may have an incentive to rate themselves too highly, but self assessment can work well if combined with additional mechanisms to discourage this[3].

We piloted pure self-assessment in an edX solid-state chemistry course in the context of learning sequences. Since the goal of problems in a learning sequences is active learning and self-monitoring mastery (as opposed to grading), students had no incentive to cheat (and were not discouraged from doing so). A TA manually graded 106 of the student submissions. 71 of the self-assessed scores were identical to the TA score. The results are shown in Fig. 1. Since the question was optional, not all students answered. As a result, there may have been substantial sample bias. Of some interest in constructing the error model for the student grader is that students very rarely grade low (and then by a very small margin).

After students finished the self-assessment step, students were given the option to enter a hint that might help their peers with the question. We have not yet analyzed this data.

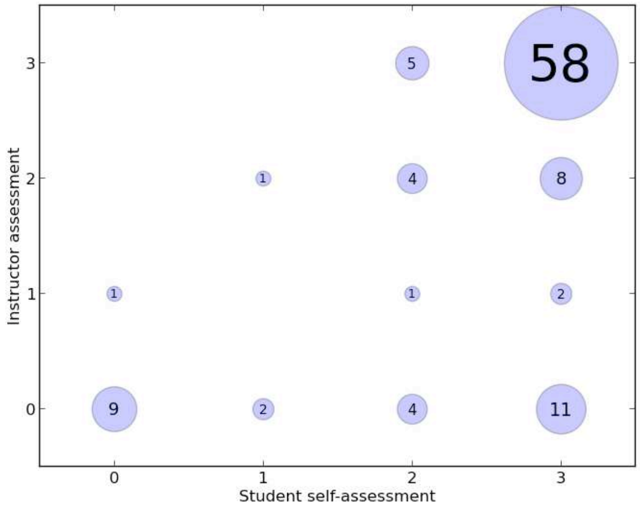

Figure 1. Accuracy for students self-assessing.

## III. Peer Assessment

In peer assessment, assignments are graded by other students who have completed the same assignment. In the edX implementation, a student first answers a question, and is then shown a set of calibration responses that were previously instructor graded and asked to grade them along defined rubric dimensions. Once he reaches an acceptable level of accuracy, he is asked to grade the responses of other students and offer feedback. Non-expert raters have been found to rival the accuracy of expert raters under the right conditions[4].

Peer assessment has been used in a variety of MOOC courses, with varying degrees of success. Klemmer [5] found that peer grading can be an extremely effective learning tool. A high percentage of students indicated that they learned more from peer-assessing the work of others than from self-assessing their own work. There was found to be a .78 Pearson correlation between self-assessed scores and peer scores. Although no data was provided on the correlation between peer-assessed scores and instructor scores, the correlation between self-assessed scores and instructor scores was found to be .91, indicating that self-assessment was an accurate scoring mechanism.

## IV. AI Grading

AI grading uses machine learning algorithms trained on instructor-scored student essays (typically, the first hundred essays submitted by MOOC students) to try to replicate instructor scoring across new essays. Once trained, AI assessment scores submissions immediately, and requires no additional human resources or input. Optionally, an instructor can rescore essays that the algorithm is not confident about (which can iteratively improve the model).

The ideas behind the algorithm used for AI assessment within edX is based on earlier work conducted during the Hewlett Foundation AES competition [6] by the VikP & jman team, where natural language processing (NLP) and machine learning techniques were used to automatically score essays.

In the Hewlett Foundation dataset, when trained with 10 fold cross validation on all available essays, it provided accuracy comparable to instructor grading. When trained on 100 essays only, accuracy falls off as expected, but is still close to instructor scoring. We tested this grading algorithm for short answers in a solid state chemistry course. Results from Hewlett, and preliminary results from chemistry, can be seen in Fig. 2.

The AI grading system does have shortcomings. It cannot reliably grade answers which do not to fit into the training examples. In some cases, it may be gameable by sufficiently clever students. It cannot give the same level of qualitative feedback as human graders. The first implementation contained feedback on spelling, grammar, and topicality. In response to student requests for more substantive feedback, rubrics have been incorporated into the AI assessment, and students can now receive feedback along an instructor-defined rubric.

The AI grading system is designed to be embedded in other platforms, and is available under an open source license. In addition, edX provides a hosted API solution.

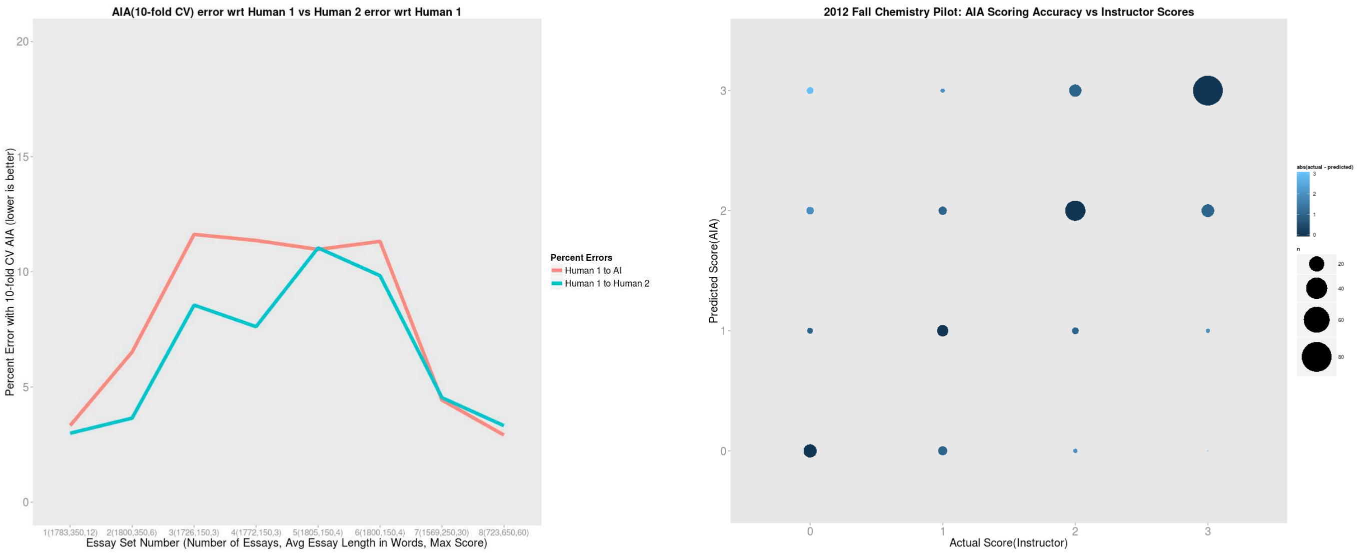

Figure 2. This plot on the left shows how well the AI grader matches a human grader on the Hewlett dataset, relative to how well two human graders match. The plot on the right compares the grades assigned by the AI grader relative to those assigned by a human instructor in the pilot chemistry course. The problems were graded by a single instructor, so we do not have numbers for inter-instructor agreement.

Question: Ion-Exchange Strengthening (3pt)
You wish to strengthen a glass by ion exchange. The initial composition of the glass is $90\%$ silicon dioxide and $10\%$ sodium oxide. Name a suitable salt bath composition with which to successfully strengthen the glass, and another one that would be unsuccessful in strengthening the glass. Explain the reasoning for your choices.

Rubric:

- Glasses fail under tension. In order to strengthen a glass, we need to create a surface compressive stress that resists an applied tensile stress.
- In order to create the surface compressive stress, we must use a salt with a larger cation that can exchange with the sodium ion in the surface layer of the glass. Potassium chloride is a possible candidate.
- Ion exchanging with a salt whose cation is smaller than sodium, such as lithium, will not create the needed compressive stresses at the surface.

Figure 3. A sample open-ended problem with rubric.

## V. Rubrics

Each problem has a rubric associated with it. A goal of the rubric is to provide a mapping from possible student answers to scores for those answers. Another goal is to identify latent traits that are required in a good answer and expose them directly to the student. A third goal is to assist in identifying and classifying student misconceptions and levels of understanding of different concepts.

For simplicity, in the edX platform, rubrics are structured among a set of instructor defined dimensions. For example, for the problem shown in Fig. 3, the rubric has three dimensions. In the current system, the rubrics are statically defined. This is in contrast to the dynamic rubrics found in the original Coursesharing system[7], as well as the hash tags in Caeser[8].

## VI. Problem Formulation

A student submits an answer $a$. In an ideal case, that answer would be matched to several positions in a rubric, one for each rubric dimension $s$. We have a set of assessment types. Each assessor $g$ has a cost $c$ associated with running it, and a cost $u$ associated with preparing it for use. In most cases, the cost $c$ is highest for instructor assessment, lowest for machine

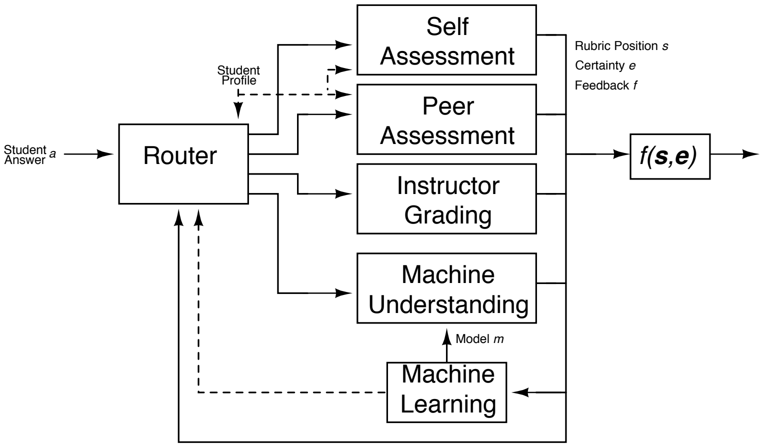

Figure 4. General formulation of routing problem.

grading, and somewhere in between for peer assessment (depending on the specific peer) and self assessment. In some cases the cost may be positive (e.g. if peer assessment is part of the teaching process). The $i$ is currently 100 for AI assessment (training data) and 20 for peer assessment (student calibration problems).

Each assessor may output the student score $\hat{s}$ on each rubric dimension, an estimate of its certainty $e$ (this may be estimated RMS error or a more complex error model), as well as qualitative feedback to the student. For each assessor, we may have an accuracy model $M_g(a)$ that estimates how likely it is to be able to generate an accurate answer.

We have several metrics we would like to optimize. First, we would like to minimize overall cost and the startup cost:

$$c = \sum_i c_i$$

$$u = \sum_i u_i$$

Second, we would like to minimize errors in grading:

$$|s - \hat{s}|$$

Finally, in a traditional explore/exploit trade off, we would like to build out better accuracy models of graders $M_g$, as well as to collect data to build better models for the AI assessment algorithm. This should lead to traditional models for judging self-assessment, such as spot checking.

## VII. Implementation

Our first implementation is based on a stepwise linear flow. It allows the instructor to define explicit work flows for how the grading is handled, with an arbitrary number of steps (practically, 5 is a reasonable maximum). For instance, an instructor could define the following work flow:

- Student self-assesses. If the self-assessment matches an instructor specified minimum and maximum score threshold (ie the student rates themselves a 2/2 or less, but above a 0/2), the student moves on to the next step.
- AI grading assesses. If the AI score matches an instructor specified minimum and maximum score threshold, the student moves on.
- Peer assessment. That score is returned, as this is the final step.

Although the steps above are specific, any type of assessment can be inserted at any step, and any thresholds between the steps can be implemented. A student is shown scores and feedback for all completed steps.

This combined approach allows instructors to use problems in a way that works for the particular domain and particular assessment goals of the problem. For example, an instructor may be interested in a flow where a student first self-assesses, and is then peer assessed, and then self assesses again in light of the feedback from peer review. An instructor interested in minimizing resource usage may only allow a student to be peer assessed after they feel that their answer is correct (ie self assess it correct). This allows us to quickly explore different variants.

In order to implement these flows, an instructor must define a rubric and a prompt. They then must commit to grading at least 20 student submissions if peer assessment is one of the steps (for student calibration), and 100 student submissions if AI assessment is one of the steps (for ML training).

## VIII. Next Steps

The edX integrated grading system provides a flexible framework for building out grading flows. In the future, flows may be tailored to a particular student via machine learning algorithms. Another potential area for improvement is in facilitating discussions between peer graders and the student who originally wrote the work. We have a mechanism for offering feedback on feedback currently that spans all assessment types, but it is preliminary. Ways to characterize students by the quality of their peer assessment or self-assessment are also potentially promising, as are annotation systems for providing feedback. We will continuously work on improving the system in general as we gather more data on how it is working in practice, and where the most pressing needs lie.

More radically, we'd like to experiment with non-traditional ways to assess students. Essays are commonly used for teaching students how to communicate, in part, due to limitations imposed by the physical classroom. Machine learning could potentially evaluate the quality of small group discussions, which may be both more pedagogically effective, and simpler technologically. In addition, we would like to allow students to upload videos and other non-traditional assessments (the system is currently limited to text and images).

## IX. Conclusion

We presented a formulation for the design of an integrated system that can assess student essays and constructed responses. The system combines self assessment, peer assessment, and AI assessment in novel, flexible, fashions. We have preliminary data about how well the system works, but as adoption increases we will have more data about the system and its learning benefits.

## References

[1] S. Thrun, D. Stavens, M. Sokolsky, I. Hwang, K. Reichelt, P. Norvig. "Introduction to Artificial Intelligence." ai-class.org.

[2] P Mitros, K. Afridi, G. Sussman, C. Terman, J. White, L. Fischer, and A. Agarwal. "Teaching Electronic Circuits Online: Lessons from MITx's 6.002x on edX." in *Proc. ISCAS*, 2013.

[3] C. Kulkarni, S. Klemmer "Learning design wisdom by augmenting physical studio critique with online self-assessment" in *Technical Report Standford*, 2010.

[4] R. Snow, B. OConnor, D. Jurafsky, A. Ng "Cheap and Fast  But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks" in *Proc. CEMNLP*, 2008.

[5] S. Klemmer http://hci.stanford.edu/publications/2012/2012-10-10-StudioCritiqueMIT.pdf

[6] The Hewlett Foundation: Automated Essay Scoring at http://www.kaggle.com/c/asap-aes

[7] A Singh. http://www.coursesharing.org

[8] M Tang. "Caesar: A Social Code Review Tool for Programming Education." *M.Eng. Thesis*, Massachusetts Institute of Technology, 2011.