

Making Static Lessons Adaptive through Crowdsourcing & Machine Learning

Joseph Jay Williams, Juho Kim, Elena Glassman, Anna Rafferty and Walter S. Lasecki.

Harvard University

Korea Advanced Institute of Science and Technology (KAIST)

MIT Computer Science and Artificial Intelligence Laboratory (CSAIL)

Carleton College

University of Michigan

Text components of digital lessons and problems are often static: they are written once and too often not improved over time. This is true for both large text components like webpages and documents as well as the small components that form the building blocks of courses: explanations, hints, examples, discussion questions/answers, emails, study tips, motivational messages. This represents a missed opportunity, since it should be technologically straightforward to enhance learning by improving text, as instructors get new ideas and data is collected about what helps learning. We describe how instructors can use recent work (Williams, Kim, Rafferty, Maldonado, Gajos, Lasecki, & Heffernan, 2016a) to make text components into adaptive resources that semi-automatically improve over time, by combining crowdsourcing methods from human computer interaction (HCI) with algorithms from statistical machine learning that use data for optimization.

INTRODUCTION

Many online education resources are arguably static and one size fits all. They provide significant content, but like a textbook, provide the same explanations and material to all users. Once the resource has been created, it generally remains constant, with text and pictures rarely changing, regardless of how helpful they are for learning. There is little technology support for instructors to do rapid content iteration, or to collect data that they can use to decide which versions are better than others.

An Intelligent Tutoring System (ITS) provides one alternative to this static instruction. ITSs deliver different versions of a learning experience to each student, based on their knowledge, attitudes, or behavior. However, this may require instructors to generate alternative versions of the content. Most instructors don't have the resources to make many different versions of lessons, such as alternative explanations and examples, and they may not be sure which explanations and examples will be most effective.

How can instructors turn static text into adaptive components that can be perpetually enhanced? This chapter discusses how the goals of instructors and of researchers in the learning sciences can be advanced by integrating crowdsourcing and design ideas from human computer interaction (HCI) with machine learning algorithms that provide automatic optimization. The approach is illustrated by describing a system that instructors can use as a plugin to improve text components of instruction. It is called the Adaptive eXplanation Improvement System (AXIS) and was first reported in Williams, Kim, Rafferty, Maldonado, Gajos, Lasecki, & Heffernan (2016a).

AXIS lets an instructor designate existing an static explanation to be turned into an adaptive, data-driven component that tests out different explanations and improves over time. Each component asks learners to explain why an answer to a problem is correct, which promotes reflection and understanding (Williams &

Lombrozo, 2010), as is well-established in education as the *self-explanation* effect (Chi et al, 1989; 1994; Lombrozo, 2006). At the same time, learners explanations can be collected and then presented to help *future* students learn, given the importance of high quality explanations (Renkl, 1997). The instructor then indicates how explanations should be compared (e.g., based on learner ratings or on learner performance on a related problem), and AXIS applies a machine learning algorithm automatically analyze and use the data being collected to choose better explanations for future students. For example, in a computer science class, one might want to determine what explanation of a sorting algorithm is most clear to learners, as measured by students' rating of the helpfulness of the explanation. If learners are provided with the opportunity to rate the explanation that was provided to them, their responses can be used to select better explanations for future learners.

The original paper (Williams et al, 2016a) provides the technical details of the system design and algorithm. The current chapter is intended to provide a higher level overview of how the approach is relevant to instructors and researchers outside of HCI and machine learning. In that vein, we begin with a brief overview of relevant work.

RELATED WORK

Human Computation & Crowdsourcing for Education

Creating usable, understandable explanations for answers to problems is a task that current artificial intelligence (AI) approaches still struggle to achieve, although great success has been made in building intelligent conversational tutors for specific tasks (e.g., Graesser, Chipman, Haynes, Olney, 2005). Crowdsourcing for human computation has been used to create useful systems in a variety of settings where AI still struggles (Doan, Ramakrishnan, & Halevy, 2011), like assistance editing word documents (Bernstein et al., 2010). These systems achieve scalability by reducing the skill level needed by individuals in a crowd to complete components of a task (e.g., see Scribe, Lasecki et al., 2012).

Paulin and Haythornthwaite (2016) explore different facets of online education that can be crowdsourced. Yuan et al. (2016) specifically explore how rubric-guided crowd workers can produce helpful feedback on students' design submissions. Weir, Kim, Gajos, & Miller (2015) introduce the idea of 'learnersourcing' as a way to elicit useful improvements to an educational system from 'crowds' of learners, as a byproduct of their natural interaction with educational content (see also Kim, 2015; Kim et al, 2014).

Glassman & Miller (2016) describe a learnersourcing system that, through personalized prompts, collects hints for students from peers who have already acquired the expertise necessary to generate them. These hints helped students debug computer processors and design better transistor-based logic gates. The current chapter is similar in collecting learnersourced explanations and hints. Additionally, our work collects feedback from students, using machine learning to analyze this data and infer how popular explanations are, then actively change the system to present explanations more frequently as evidence accrues as to their effectiveness.

Tradeoffs between Experimentation & Optimization

Randomized experimental comparisons are a powerful method for quantitatively determining what is effective – from instructors and researchers identifying which of multiple explanations is satisfying, to doctors identifying which drug promotes health to product designers evaluating new interfaces. Experiments have brought great value to web analytics under the label of *A/B testing* (Kohavi, Longbotham, Sommerfield, & Henne, 2009).

Typically experimental comparisons assign people to conditions with equal probability (e.g., 50% to explanation A and 50% to explanation B) since this provides maximal statistical power to detect differences, relative to random assignment that uses non-equal probabilities (e.g., 30% to explanation A and 70% to explanation B). Both approaches are random in the sense that it is not known *which* condition a participant will be assigned to (that is sampled), but not random in the more limited sense that the probability of being assigned to any condition is equal. Both approaches are statistically valid in allowing causal conclusions. The approach of random assignment with different probabilities provides the opportunity to reduce (but not eliminate) statistical power to discover differences, in order to maximize *actual benefit* to the participants in the experiment, since a greater proportion of them could receive the more beneficial condition.

This raises many instructional, statistical, and ethical questions, which instructors and researchers will themselves have to answer. In this chapter we describe how this connects to a classic problem in reinforcement learning: Balancing exploitation of what has already been observed with exploration to learn more about different options.

Reinforcement Learning: Multi-Armed Bandits

Reinforcement learning (RL) is a type of machine learning that allows a system to learn through interaction with an environment (see Sutton & Barto, 1998 for an overview). RL algorithms try out different actions, and seek to determine what actions tend to be most effective. What is “effective” is defined by the algorithm designer as a particular parameter to optimize; in the case above, an effective website is one that generates mailing list sign ups.

In cases like testing different versions of websites or deciding what ads to show users, a class of algorithms known as *multi-armed bandit* algorithms are commonly used. The term multi-armed bandit comes from analogy to slot machines (‘bandits’ that ‘steal money’ when you pay them) that have multiple arms to pull, with unknown and uncertain payoffs. A player may be trying to determine what arm to pull (what action to take) to get as large of a payoff as possible, which requires balancing exploration of arms against exploitation of arms that seem to be good.

While there are many different bandit algorithms (e.g., Auer, Cesa-Bianchi, & Fischer, 2002; Chapelle & Li, 2011), all of them seek to optimize the total effectiveness of the selected options, and do so even when effectiveness is stochastic. In the case of educational applications, we define the payoff based on some measure we are trying to optimize, such as learning or motivation, and the “arms” (actions) are different versions of educational content. Multi-armed bandit algorithms provide a scalable, model free way of

optimizing our choices so that we will end up being as effective as possible for as many students as possible (Clement et al, 2014; Liu, Mandel, Brunskill, & Popovic, 2014).

Multi-armed bandit algorithms seek to maximize the total cumulative effectiveness of all actions over a period of time. For instance, if after one hour, 2% of users of version A and 5% of users of version B have signed up for the website, this type of algorithm will tend to assign more users to version B, since it has observed evidence that this version is more effective. However, it will still allocate some users to version A because it is not yet certain which is better: it may just be by chance that more people have signed up with version B, especially if only a small number of users have viewed the site. As the number of users increases, the algorithm's confidence in which version will be more effective increases, and it will tend to assign almost all users to the most effective option.

MAKING STATIC TEXT COMPONENTS ADAPTIVE

We illustrate our approach in a system for creating adaptive explanations for math problems, the Adaptive eXplanation Improvement System (AXIS; Williams et al., 2016). In the following sections we provide context about the value of explanations, then describe how an instructor can set up AXIS for adding and adapting the explanations for how to solve math problems.

Explanations in Math Problems

To provide context for an instructor's goal of making explanations in math problems adaptive, consider the ubiquitous structure of problem activities in platforms like Khan Academy, ASSISTments, EdX, and Coursera. Students attempt a problem and are given feedback about whether their answer is correct. They may also be able to request an explanation or elaboration that explains why an answer is correct, or how to solve the problem. The value of providing quality explanations is clear from the educational and psychological literature, although it is an active area of research as to which explanations are deemed satisfying, and actually help learning (Lombrozo, 2006; Renkl, 1997).

Learnersourcing Interface for Eliciting Contributions

Typically, a student might see no explanation after solving a problem, or if one is given, it was written by the instructor who wrote the problem. To elicit explanations from *students* that can be used for future learners, we add a reflection prompt, asking the student to explain why the answer was correct in their own words. The prompt tells the learner that generating the explanation will help them solidify their knowledge, as shown in Figure 1. The design of this prompt is guided by decades of research in psychology and education on the benefits of generating explanations for learning (Chi et al., 1994; Williams & Lombrozo, 2010; Williams, Lombrozo, Hsu, Huber, & Kim, 2016).

The prompt of course is designed to serve a second function, which is to generate explanations that future students might find helpful, in the spirit of crowdsourcing in HCI. These learner-generated explanations can be tested on future learners using reinforcement learning algorithms that evaluate the explanations using performance metrics designated by instructors.

Explain out loud and in your own words how to solve the problem. Then write the explanation below.

You have probably heard of the saying "the best way to learn is to teach".

Right now, try **explaining out loud** why the answer above is correct, and how to solve the problem. Imagine explaining to another learner, if the two of you were sitting at your computer working on this together.

Then write your explanation into the text box below. It will help you, and could help another learner similar to you.

You will might feel as though you don't understand this well enough to explain it. But constructing an explanation will **still** help you learn, by helping you spot gaps in your knowledge, and connecting different facts and principles together.

Explaining will prepare you better for the problems that are coming up.

As you write, you can create a helpful explanation by copy/pasting some text from the explanations you received. But don't do this without making changes if these aren't the words you would actually use.

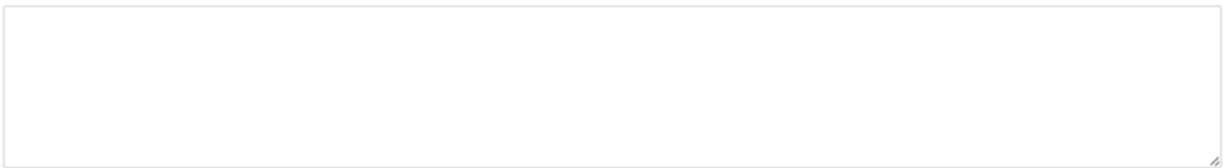


Figure 1. A prompt for a learner to explain how to solve a math problem.

Choosing Performance Metrics

The instructor must decide what metric the system should be maximizing via its choice of text component. For example, in the Williams et al. (2016a) deployment of AXIS, we decided to maximize learners' ratings of how helpful explanations were for learning. To maximize learners' ratings of helpfulness, an instructor can insert an additional question for learners who receive an explanation to answer: "How helpful do you think this explanation is for learning?" Students can respond on a Likert scale, such as: 1 (Absolutely Unhelpful) to 10 (Perfect), as shown in Figure 2. In future versions of this system, instructors could choose to optimize a *combination* of performance measures, such as learners' accuracy on subsequent problems and the likelihood that they keep working.

Explanation: Here is an explanation someone wrote of why the answer is right, and how to solve the problem.

The probability of getting a chocolate cookie on his first draw is $5/8$. If he draws a chocolate cookie, there will be 4 chocolate cookies and 3 oatmeal cookies left, so the probability of getting an oatmeal cookie on his second draw is $3/7$. $(5/8) \cdot (3/7) = 15/56$.

How helpful do you think this explanation is for learning?

Absolutely Unhelpful										Perfect
1	2	3	4	5	6	7	8	9	10	
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Figure 2. An example of an explanation of how to solve a particular math problem, with a prompt to rate the helpfulness of the explanation.

An instructor might wish to optimize a more direct measure of learning or progress toward the instructor's long-term goals. For instance, the value of each text option could be based on how many learners who

saw that option got the next problem correct. We considered this measure but found that confounding factors, such as variations in knowledge across students, overwhelmed the relative differences in quality between explanations. Increasing the number of learners using the system would allow the machine learning algorithms to better cope with these confounding factors.

Deployment of AXIS

We (Williams et al, 2016a) deployed the AXIS system with 75 and then 150 learners, numbers comparable to multiple offerings of large residential introductory courses, and sought to optimize the helpfulness of the explanations as rated by learners.

As learners solved problems, their construction of a deep understanding was guided by prompting them to explain why answers were correct. These explanations were captured and then provided to *other* learners, so that after receiving the correct answer, future learners could read an explanation of why the answer was correct. Learners were asked to indicate how helpful an explanation was on a scale from 1 to 10. A multi-armed bandit algorithm then automatically incorporated the rating from each new learner, and used the ratings to alter the probabilities of which explanation it would present to the next learner. Those that tended to be highly rated were presented more frequently, and AXIS automatically incorporated new explanations that were given by users into the pool of explanations that it delivered. Figure 3 shows examples of explanations from several different categories: explanations discarded by AXIS, explanations identified as good by “Early Stage AXIS” (after just 75 learners), explanations identified as good by “Later Stage AXIS” (after 150 learners), and explanations written by an instructional designer on the ASSISTments platform.

We evaluated the explanations produced by the AXIS system after 75 learners (“Early Stage AXIS”) and 150 learners (“Later Stage AXIS”). Relative to the original math problems that had no explanations, adding AXIS’s crowdsourced and intelligently selected explanations to math problems significantly improved learners’ subjective learning experience, as measured by their ratings of how much they learned (effect size of AXIS explanations impact on subjective ratings of learning, vs no explanation: $d = 0.29$; vs instructional designer: $d = 0.06$; vs filtered explanation: $d = 0.22$).

In addition, these learner-generated explanations from AXIS also objectively increased learning, as measured by performance improvements on related problems that were presented in a *pre-test* (before studying problems) and then again in a *post-test* (after studying problems under different conditions, like receiving vs not receiving AXIS explanations). Overall increases in accuracy solving problems were observed for AXIS explanations, vs no explanation : $d = 0.19$; vs instructional designer: $d = 0.09$; vs filtered explanation $d = 0.01$. Breaking up accuracy into *isomorphic* and *transfer* problems: The increases were observed on *isomorphic* problems, that were identical except for changing the numbers in the problem (vs no explanation: $d = 0.12$; vs. instructional designer: $d = 0.11$; vs. filtered explanation: $d = 0.07$), as well as on *transfer* problems, that were completely novel but tested generalization of the concept (vs no explanation: $d = 0.22$; vs instructional designer: $d = 0.04$; vs filtered explanation: $d = 0.26$).

	Explanation	Explanation Rating
Learner Explanation AXIS Discarded via Filtering Rule	It is three over seven because after the chocolate cookie has been removed there are 7 cookies in the jar, leaving 3 oatmeal cookies remaining.	5.2
Early Stage AXIS	go based on the amount of cookies that are available and run a trial until the chocolate cookie is picked out, then do the same for oatmeal	4.2
Later Stage AXIS	When you have 8 cookies in the jar and 5 are chocolate you have a $\frac{5}{8}$ chance of the cookie you draw being chocolate. When there are 7 cookies in the jar and 3 are oatmeal you have a $\frac{3}{7}$ chance of drawing the oatmeal cookie. To get the overall probability you need to multiply $\frac{5}{8}$ by $\frac{3}{7}$ which results in overall probability of $\frac{15}{56}$	6.8
Written by Instructional Designer	The total number of cookies in the jar is 8. Since there are 5 chocolate cookies the probability that Chris gets an chocolate cookie is $\frac{5}{8}$ Since Chris removed 1 cookie from the jar and did not replace it or put it back there are now 7 cookies in the jar. So, the probability that Chris gets an oatmeal cookie from the jar is $\frac{3}{7}$ $\frac{5}{8} \times \frac{3}{7} = \frac{15}{56}$ So, the probability of Chris getting a chocolate cookie on the first draw, and an oatmeal cookie on the second draw is $\frac{15}{56}$ Type in $\frac{15}{56}$	7.7

Figure 3. Examples of learner-generated explanations collected and delivered by AXIS, as described in Williams et al., 2016.

Instructor Review in AXIS

At any point, the instructor can inspect the pool of explanations in the system, see the policy for how often each explanation is being presented, and also see the data about how explanations have been rated so far by students. Figure 4 shows a screenshot from a prototype view we have built for instructors, which provides a rough illustration of the kind of information instructors can see, and how they could change components of AXIS. This can be valuable in informing instructors about their “expert blind spots” about which explanations students will find helpful (Nathan, Koedinger, Alibali, 2001). The policy is the probability that each explanation will be presented, and directly reflects the reinforcement learning algorithm’s “judgment” of how much evidence suggests that this explanation is the best explanation, with respect to the instructor’s performance metric.

	A	B	C	D	E	F
1	Version	Explanation	Instructor "Priors"	Student Ratings	Mean Rating	Probability of Presentation
2		Every number has one other number that will sum it to 8. After the first spin there's a 20% chance (1 out of the 5 options) that the wheel will land on the one number that sums the first number to 8	9 10	6 8 6 7	6.75	30%
3		There's a 1 in 5 chance because there are 5 choices and each trial is independent of the other.	9 10	3 2	2.5	15%
4		Ok given any first spin, there is one and only one number on the wheel which when added to it will result in 8. Since there is one number which works and 5 that do not, the chance is 1 in 5.	9 10	9 8 9 10	9	55%
5						0%
6						0%

Figure 4. Illustration of a rough prototype for how instructors can view AXIS explanations, data, and policy.

The design of the system also allows the instructors to interact with AXIS, although Williams et al (2016a) did not explore this. The AXIS front-end interface pulls the actual explanations from the Google sheet in Figure 4, computing the policy based on the data in that Google Sheet, using the probabilities displayed. Any modifications to that sheet directly impact the behavior of the AXIS system. This provides a range of promising functionality, that needs to be carefully studied.

The instructor can directly add their own explanations at any point in time, by typing them into the sheet. AXIS will then start presenting it alongside student explanations. Instructors can also directly increase or decrease the probability of an explanation being presented. They could do this by directly overwriting the Probability of Presentation in column F, although this will override the AXIS algorithm's computation of a policy. A way for the instructor to *interact* with rather than *override* the AXIS algorithm would be by giving the explanation their *own* rating of how helpful it is for learning, entered into column C. Using a Bayesian approach that exploits the conjugate distributions in Thompson Sampling, the instructors could encode their beliefs about the quality of the explanation as giving a certain number of 'fictional' observations of student ratings. In Williams et al (2016a) this was not exploited, since we assumed that the "prior" for every explanation was just two fictional observations, as if one student had rated it a 9 and another a 10.

Future works needs to explore how to elicit these judgments from instructors in a way that is effectively incorporated into the algorithm. For example, this requires specifying how much weight should be given to the instructor's rating of the explanation relative to an individual learner, which could be based on their confidence in their rating. Such capability can help instructors fine-tune the system when learner-generated explanations contain misconceptions or the instructor wants to direct learners to a particular explanation. The general approach we use aims to empower instructors to build adaptive educational systems, in the spirit of end-user programming that allows people to make changes to software systems without writing code (Myers, 1995).

Implementation & Use of AXIS: Available Resources & MOOClet Framework

While the AXIS components can be implemented using any technology of choice by the instructor team, we have provided an online tool to make it easy for instructors to use AXIS in their own classes. To find out more, sign up at the URL <http://tiny.cc/useaxis>. The tool allows instructors to submit their own problems and questions to get a version of AXIS for their own demo or classroom. Their version of AXIS can then be provided directly to students (e.g., by emailing a URL) or embedded into a range of MOOC platforms and learning management systems, like EdX, Canvas, Moodle, since these and AXIS both use the Learning Technologies Interoperability (LTI) standard.

The crowdsourcing and machine learning approach used in AXIS can be applied to adapting a wide range of other resources. For example, an identical approach has been applied to adapting and personalizing which motivational emails students receive, which study tips appear above problems, and which lesson pages are presented. The technology and approach that support this broad range of adaptation is described by the MOOClet framework (Williams & Heffernan, 2015 and; Williams, Kim, Li, Whitehill, Maldonado, Pechenizky, Heffernan, 2014). This framework allows any existing website, app, or online resource to leverage crowdsourcing and machine learning to produce rapid adaptation, providing data in real-time to both instructors and algorithms immediately after students interact with resources.

SUMMARY, RECOMMENDATIONS AND FUTURE WORK

In this chapter, we have described a systematic approach for transforming static lessons into adaptive resources, in the context of self-improving explanations for how to solve mathematics problems. By drawing on research on explanation from psychology and education, we designed prompts that enhanced the learning of both the students who generated the explanations and the future students who receive the explanations. The system was designed by combining crowdsourcing principles to elicit learner responses at scale, and statistical machine learning algorithms to optimize which explanations are presented, using target metrics defined by instructors. Instructors and researchers who wish to use this system or learn more can sign up at <http://tiny.cc/useaxis>. We briefly discuss future applications and extensions of this type of system.

Beyond Math Problems: Adapting Hints and Discussion Forum Q&A

Many features of a learning environment can be enhanced by iteratively improving written text or recommendations. Massive Open Online Course (MOOC) discussion forums include many examples of questions asked and answered by learners and course staff. Our system could recommend answers, based

on the objective performance or subjective ratings of the students who accessed them. In addition to providing explanations *after* an answer is known, this system could adaptively provide *hints* for how to solve problems. The current system could be used to select and optimize the distribution of learner-generated hints from a learnersourcing system like Glassman & Miller's (2016) in real-time.

Personalization

The current approach is limited by a one-size-fits-all assumption: that there is a single best-explanation for everyone, in every context, independent of their learning goal. An alternative is to collect data about differences between individuals' context and characteristics, and use this in learning *personalized* policies. When given individual's context and characteristics, reinforcement learning algorithms can also learn a policy for **whom** to deliver **which** content to. Learning personalized policies does not require without creating new explanations or modifying existing ones. The existing approach can be straightforwardly extended to create adaptive components that become personalized, once there is a metric for differences between learners.

Summary

Our system allows an instructor to launch with a single explanation, hint, or answer, and leave open the option to *improve* this over time, as data becomes available about what works for learners. The system is easily extendable to tailor explanations to individual learners based on their specific gaps in knowledge or misconceptions. Frameworks like our system provide the opportunity for an entire community of instructors and learners to be involved in instructional design, treating the activity as an ongoing resource to be improved and refined.

For example, if lessons, discussions, or problems in a single MOOC are used by a wide variety of classes, any instructor from these classes can contribute text components, as well as expert opinions on which text components are effective. Rather than making hard, permanent decisions about what explanation to present, the system probabilistically combines opinions and evidence from many different sources. Future ITS can benefit from incorporating these adaptive text components into existing lessons. Future work focused on personalizing explanations based on the characteristics of individual learners and testing the effectiveness of these systems in a larger set of domains will enable ITS designers to better understand how these systems might be of the most benefit for increasing learning.

REFERENCES:

- Aleven, V. A. and Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive science*, 26(2):147–179.
- Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3), 235-256.
- Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., & Panovich, K. (2010). Soylent: a word processor with a crowd inside. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology* (pp. 313-322). ACM.
- Chi, M. T., Bassok, M., Lewis, M. W., Reimann, P., and Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive science*, 13(2):145–182.

- Chi, M.T.H., de Leeuw, N., Chiu, M.H., LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, 18, 439-477.
- Doan, A., Ramakrishnan, R., & Halevy, A. Y. (2011). Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4), 86-96.
- Clement, B., Oudeyer, P.-Y., Roy, D., and Lopes, M. (2014). Online optimization of teaching sequences with multi-armed bandits. In *Educational Data Mining 2014*.
- Kim, J. (2015). *Learnersourcing: Improving Learning with Collective Learner Activity*. PhD thesis, Massachusetts Institute of Technology.
- Kim, J., Guo, P. J., Cai, C. J., Li, S.-W. D., Gajos, K. Z., and Miller, R. C. (2014). Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 563–572.
- Glassman, E. L. and Miller, R. C. (2016). Leveraging Learners for Teaching Programming and Hardware Design at Scale. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work & Social Computing* (pp. 37-40). ACM.
- Graesser, A. C., Chipman, P., Haynes, B. C., and Olney, A. (2005). Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Learning Technologies*. 48(4):612–618.
- Weir, S., Kim, J., Gajos, K. Z., & Miller, R. C. (2015). Learnersourcing subgoal labels for how-to videos. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (pp. 405-416). ACM.
- Kohavi, R., Longbotham, R., Sommerfield, D., & Henne, R. M. (2009). Controlled experiments on the web: survey and practical guide. *Data mining and knowledge discovery*, 18(1), 140-181.
- Lasecki, W. S., Miller, C., Sadilek, A., Abumoussa, A., Borrello, D., Kushalnagar, R., & Bigham, J. (2012, October). Real-time captioning by groups of non-experts. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (pp. 23-34).
- Liu, Y. E., Mandel, T., Brunskill, E., & Popovic, Z. (2014, July). Trading Off Scientific Knowledge and User Learning with Multi-Armed Bandits. In *Educational Data Mining*.
- Lombrozo, T. (2006). The structure and function of explanations. *Trends in cognitive sciences*, 10(10):464– 470.
- Myers, B. A. (1995). User interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(1):64–103.
- Nathan, M. J., Koedinger, K. R., and Alibali, M. W. (2001). Expert blind spot: When content knowledge eclipses pedagogical content knowledge. In *Proceedings of the Third International Conference on Cognitive Science*, pages 644–648. Citeseer.
- Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. *Cognitive science*, 21(1):1–29.
- Olney, A. M., Brawner, K., Pavlik, P., & Koedinger, K. R. (2015). Emerging Trends in Automated Authoring. Design recommendations for adaptive intelligent tutoring systems: Learner modeling, Vol. 3 of Adaptive Tutoring (pp. 227-242). Orlando: U.S. Army Research Laboratory.
- Olney, A. M., & Cade, W. L. (2015). Authoring Intelligent Tutoring Systems Using Human Computation: Designing for Intrinsic Motivation. In D. D. Schmorow & C. M. Fidopiastis (Eds.), *Foundations of Augmented Cognition* (Vol. 9183, pp. 628–639). Springer International Publishing.
- Malone, T. W., & Bernstein, M. S. (2015). *Handbook of Collective Intelligence*. MIT Press.
- Paulin, D. & Haythornthwaite, C. (2016). Crowdsourcing the curriculum: Redefining e-learning practices through peer-generated approaches. In *The Information Society*. 32(2), 130-142.
- Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction. *Cambridge: MIT press*.
- Williams, J. J., Kim, J., Rafferty, A., Maldonado, S., Gajos, K., Lasecki, W., & Heffernan, N. (2016a). AXIS: Generating Explanations at Scale with Learnersourcing and Machine Learning. *Proceedings of the Third Annual ACM Conference on Learning at Scale*.
- Williams, J. J., Hsu, A., Huber, B., Kim, J. (2016b). Revising Learner Misconceptions Without Feedback: Prompting for Reflection on Anomalous Facts. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*.
- Yuan, A., Luther, K., Krause, M., ICSI, U., Vennix, S., Dow, S. P., & Hartmann, B. (2016). Almost an Expert: The Effects of Rubrics and Expertise on Perceived Value of Crowdsourced Design Critiques. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work & Social Computing* (pp. 1005-1017). ACM.