

# Learnersourced Recommendations for Remediation

Shang-Wen “Daniel” Li  
MIT, Cambridge, MA  
swli@mit.edu

Piotr Mitros  
edX, Cambridge, MA  
pmitros@edx.org

**Abstract**—Rapid remediation of student misconceptions and knowledge gaps is one of the most effective ways to help students learn [1]. We present a system for recommending additional resources, such as videos, reading materials, and web pages for students working through on-line course materials. This can provide remediations of knowledge gaps involving complex concepts. The system relies on learners suggesting resources which helped them, leveraging economies of scale as found in MOOCs and similar at-scale settings in order to build a rich body of remediations. The system allows for remediation of much deeper knowledge gaps than in prior work on remediation in MOOCs. We validated the system through a deployment in an introductory computer science MOOC. We found it lead to more in-depth remediation than prior strategies.

## I. INTRODUCTION

In learning at scale systems (e.g. MOOCs, shared SPOCs, etc.) resources are created once, instead of thousands of times, allowing for better quality at lower total cost. Still, even one-time creation of high-quality resources and implementation of evidence-based pedagogies is expensive, especially moving into the diversity of courses offered at university level. For example, intelligent tutoring systems (ITSs) have shown that keeping students in a state of engagement between boredom and confusion (known as constructive struggle[12] or flow[2]) can have a substantial positive impact on learning[13]. Unfortunately, such systems are cost-prohibitive to implement at a scale of thousands of courses. We believe crowdsourcing from learners is an economical and practical solution to create tools for remediation.

Previous crowdsourced work focused on the creation of remediations for common incorrect student answers [10]. For example, students who arrived at an incorrect answer and later a correct answer could submit a remediation which would be seen by future students who made the same mistake. This is similar to several industry ITS systems such as MasteringPhysics, but at much lower content creation costs. Besides, both experts and novices have well-researched limitations[6][5][11]. By combining both expert and novice contributions, such framework can provide remediations with much greater breadth and make for more effective teaching and learning.

While helpful, this is limited in the scope of problems it can be applied to, since it requires students to arrive at an answer before providing a remediation. In classroom observations of STEM courses, most students get stuck before reaching an answer. This limits the domain of applicability to relatively narrow, simple questions. Complex questions, such as design projects, or even multiconcept physics questions, would fall outside of the scope such systems.

Students also have access to forums. Forums act as both a place students can ask questions, and a repository of remediations to common student misconceptions[9]. There are still several gaps, key of which is that students learn better when first given the opportunity to struggle, and solve problems with the minimum level of scaffolding required to succeed[7]. Most MOOC forum posts target specific errors, and provide more scaffolding than desired for a first intervention.

In this paper, we describe a system which provides students with a list of remediation resources in the form of pre-existing web pages. By limiting resources to pre-existing web pages, we limit remediations to broad ones, such as explaining relevant concepts, rather than ones focused on a specific bug. Students are given a set of tools for jointly managing these resources, and can suggest new resources, edit descriptions of those resources, up/down voting resources, and flagging spam and abuse. This system allows stuck students to self-navigate to remediations which can provide additional information.

We deployed this system on the edX platform as an XBlock in a MOOC in introductory computer science, making this one of the largest deployments of an educational recommender system to-date. Students effectively contributed a body of resources, and those resources, indeed, had more depth than traditional MOOC forums. We did not conduct a randomized control trail, so measuring the effect of these resources on student learning remains future work.

## II. SYSTEM OVERVIEW, DESIGN AND RESULTS

Fig. 1 gives an overview of the user interface, which is designed to leverage existing user knowledge by building on affordances from established platforms, such as reddit, as well as portions of the edX platform. We tested the Recommender in an eight-week computer science MOOC offered on edX (“6.00.1x Introduction to Computer Science and Programming Using Python” from the Massachusetts Institute of Technology). This course has weekly problem sets. Every problem set has an overarching theme or narrative. We attached a Recommender to each page of a problem set.

Since the goal of the Recommender is to provide interventions with more depth and less scaffolding than forums or simple hinting systems, to evaluate its effectiveness, we manually tagged all of the recommended resources, as well as 100 discussion forum threads sampled at random into one of four categories: 1) *Shallow*: understanding the text of the problem, programming language syntax, terminology, error messages, simple debugging, etc., 2) *Medium*: understanding the concepts required to solve the problem, summarizing, and processing the ideas learned in problem, 3) *Deep*: extending on the materials taught in the course (e.g., suggesting a better

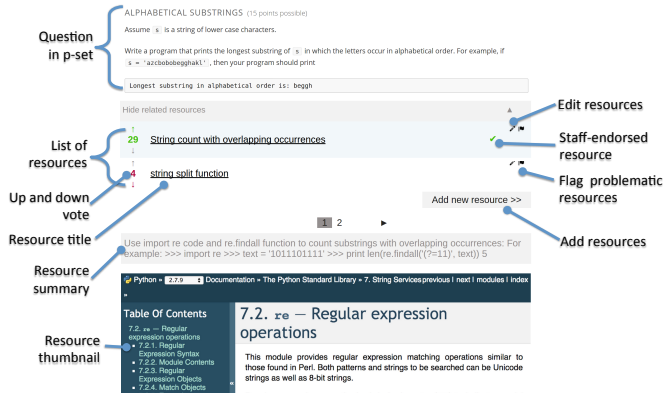


Fig. 1. An overview of the Recommender, the crowdsourcing point-of-need help system described in this paper. The system is typically placed below a problem in a homework assignment, and displays a list of links to resources. Each resource includes a title, and several means for resources manipulation. Below the resource list, there is a description of that resource shown on mouse-over. Typically, this consists of text explaining what the resource is and when and how it might be helpful, optionally together with an image, such as a screenshot. Students have tools to upvote and downvote resources, and to flag inappropriate resources. Staff have additional controls to remove bad resources and endorse especially helpful ones. In addition, there are controls for creating new resources, and for browsing through multiple pages.

technique for solving a problem than that presented in the course), and 4) Other: forum conversation, off-topic, etc. The results, shown in Fig. 2, showed that forums and the Recommender system were complementary. Learners recommended general, high-level resources, such as pointers to documentation, libraries, and blog posts which expanded on the content in the course, in some cases, allowing more elegant solutions to the problems. In contrast, forums provided high quality remediations for debugging specific simple issues, such as understanding problem text or basic debugging.

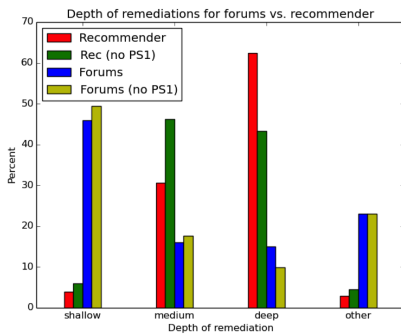


Fig. 2. Depth of remediations from forums vs. from the Recommender, classified as described in the text. We report results with and without the first problem set, where results were skewed by a problem which involves installing the development tools. This problem received a disproportionately large number of recommendations, and biased the results.

### III. CONCLUSION

In this paper, we showed a way in which students could collaborate around improving resources for remediation. The course had 13,225 active students in the run where we performed most of our analysis, and we saw 102 contributed resources over 7 problem sets, and sufficient voting activity to organize those resources. 23% of the students who accessed

the problem set also accessed the Recommender. Qualitatively, students contributed useful resources for virtually all of the places the Recommender was used where there were relevant resources. Unsurprisingly, we found several places, such as simple one-step problems, where recommended resources were not relevant, and indeed, students did not contribute resources at all. We were encouraged by these results. The Recommender is a remediation system targeted at students who need additional help, with a goal of maximizing quality (rather than number) of contributions.

The results also support our thesis that successful remediation of a range of student issues requires a range of techniques, depending on the type of issue. The Recommender is helpful for complex activities, such as design problems, and complements other remediations techniques, such as community Q+A and simple hinting, which are helpful in simpler contexts.

As next steps, we are working to integrate better on-ramps and discoverability into the system. It seems likely that clear guidance, instruction, and advertising could improve usage and contribution levels. In addition, we would like to make the experience more collaborative. Students should be able to discuss recommended resources, receive attribution for contribution, and see the positive impact of their contributions on fellow students. Besides, we are considering a range of machine-assisted techniques for helping more efficiently manage the quality of the resources. Lastly, a randomized trial ought to be performed to evaluate the effect of Recommender on student performance.

The source code and documentation for the system is available under an open source license at <https://github.com/pmitros/RecommenderXBlock>.

### REFERENCES

- [1] J Anderson, A Corbett, K Koedinger, and R Pelletier. Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 1995.
- [2] Mihaly Csikszentmihalyi. *The Psychology of Optimal Experience*. Harper and Row, 1990.
- [3] Hendrik Drachsler, Katrien Verbert, Olga C. Santos, and Nikos Manouselis. Panorama of recommender systems to support learning. (*Pre-publication*), 2014.
- [4] Mojisola Erdt and Christoph Rensing. Evaluating recommender algorithms for learning using crowdsourcing. *Int. Conf. on Advanced Learning Technologies*, 2014.
- [5] D Feldon. The implications of research on expertise for curriculum and pedagogy. *Educ Psychol Rev*, 2007.
- [6] P Hinds, M Patterson, and J Pfeffer. Bothered by abstraction: The effect of expertise on knowledge transfer and subsequent novice performance. *Journal of Applied Psychology*, 2001.
- [7] K Koedinger and V Aleven. Cognitive tutors: Lessons learned. *Educ Psychol Rev*, 2007.
- [8] N Manouselis, H Drachsler, K Verbert, and E Duval. *Recommender Systems for Learning*. Springer, 2013.
- [9] P Mitros, K Affidi, G Sussman, C Terman, J White, L Fischer, and A Agarwal. Teaching electronic circuits online: Lessons from MITx's 6.002x on edX. In *ISCAS*, pages 2763–2766. IEEE, 2013.
- [10] P Mitros and F Sun. Creating educational resources at scale. *IEEE Conf. on Adv. Learning Technologies*, 2014.
- [11] National Research Council. *How People Learn*, pages 31–50. National Academy Press, 2000.
- [12] C Seeley. *Faster Isn't Smarter*, pages 88–92. MSP, 2009.
- [13] K VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.