Improvement on Hint and Explanation Crowdsourcing Method for an Online Learning Platform

by
Thanapron Patikorn

A Dissertation
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in
Computer Science

by

April 28, 2021

APPROVED:

Professor Neil Heffernan, Advisor

Professor Joseph Beck, Committee Member

Professor Jacob Whitehill, Committee Member

Professor Adam Sales, External Committee Member

Professor Craig Wills, Head of Department

Contents

Acknowledgement

First and foremost, I am deeply grateful to my advisor Professor Neil Heffernan for the opportunity to work with the ASSISTments project as well as for introducing me to many wonderful collaborators from many different places. I can't begin to describe how much I've learned from my many years working with him and ASSISTments.

I would like to thank my committee members Professor Joseph Beck, Professor Jacob Whitehill, and Professor Adam Sales. I've learned many lessons, both theoretical and practical, from many classes and projects I've worked with each of them. It's not exaggerated to say that I now see the world differently because of what I learned from them.

I would like to also thank fellow students working under Professor Neil Heffernan as well as staff members at ASSISTments. Without their support, none of my projects would be possible. Special thanks to Dr. Douglas Selent who had helped me greatly when I first started my Ph.D. program and whose work, PeerASSIST, greatly shaped my research work during my time as a Ph.D. student.

I would like to extend my sincere thanks to my fellow students of the Thai Scholarship Program. Their continuous friendship and support have pushed me through countless hardships I've faced in this journey.

And, last but not least, I would like to offer a special thanks to my wonderful family: my mother Supetcharat Patikorn for always supporting my choices, my father Sataporn Patikorn for igniting my love of math, and my sister Patchraporn Patikorn for always being there when I need someone.

Abstract

Crowdsourcing has been used in many successful online applications such as Wikipedia and Stack Overflow. In the field of educational research, many educational platforms, such as edX, recently implemented features that improve learning by taking advantage of crowdsourcing, such as peer grading.

In my previous work, I implemented a crowdsourcing feature called "TeacherASSIST" inside the ASSISTments online learning platform. TeacherASSIST allowed teachers to create hints and explanations, which would be given to students on-demand while they were working on their assignments. In that work, I used a simple aggregation method that automatically distributed such hints and explanations created by expert-selected teachers to all students inside ASSISTments. By using randomized controlled trials, I found that crowdsourced hints and explanations improved student learning with statistical significance.

In this work, I improved TeacherASSIST by adding more organic approaches to aggregate hints and explanations using trusted teachers and hint/explanation ratings. The first approach was allowing teachers to designate other teachers as 'trusted.' Their students would then be able to receive hints and explanations created by the trusted teachers. The second approach was constructing global teacher scores based on teachers rating each other's hints and explanations. The aggregation based on ranking allowed the pool of globally trusted teachers to grow over time even for teachers who do not actively search for more trusted teachers. I then ran a randomized controlled trial and used linear regression models to evaluate the effectiveness of the aggregation method based on ranking. In addition, I also designed and generated prototypes of reports that would allow teachers to see how much their student supports have helped their students and, for starred teachers, other students inside ASSISTments.

Chapter 1

Background

In recent years, online learning platforms and massive open online courses (MOOCs) have gained tremendous popularity since internet access became more accessible and less expensive. In addition, teachers can now connect to other teachers teaching the same topics, allowing them to share and improve each other's resources with ease. Such a bottom-up, open, creative process where a task is completed by the people or users, partially or entirely, and then aggregated into solutions that they can all use is called "crowdsourcing" [8].

Crowdsourcing has been used in many successful applications. When talking about crowdsourcing platforms, it's impossible to not think of Wikipedia. Wikipedia is a crowdsourced, free, online encyclopedia. Wikipedia volunteer writers ("Wikipedians") consist of people of a wide range of specialties, languages, and countries of origin, allowing Wikipedia to have wide ranges of articles, written in many different languages, and can be updated in real time. However, the crowdsourcing nature of Wikipedia also causes several issues such as vandalism, racial, gender, and political bias.

Another successful example of a crowdsourcing platform is Stack Overflow which is a question-and-answer website for programmers, from learner-level to professional-level. Stack Overflow crowdsources both questions and answers from its users, allowing it to gather a large and diverse set of both questions and answers. Crowdsourced answers have advantages that they're not limited to the knowledge of a few experts employed by the site using specific programming languages and development environments.

1.1 Stages of Crowdsourcing

The process of crowdsourcing can generally be described in 4 stages [18]:

The first stage is pre-selection of contributors. During this stage the system or the organizer decides who could be contributors. For instance, anyone can edit Wikipedia articles. However, for contested, controversial articles, direct edits could be disabled for the general public, and only Wikipedians agree on certain changes through discussion that the changes are applied to the articles. For Stack Overflow, anyone could post to ask and answer a question.

The second stage is accessibility of peer contributions. During this stage, users (contributors and non-contributingusers) may access other users' contributions, which could have varying degrees of accessibility, from full access, to read-only, and no access at all. For instance, Wikipedia allows for full access to everyone's contribution. In contrast, Stack Overflow only allows regular users to view other users' contributions. However, users with enough reputation points are allowed to edit any questions and answers, including ones posted by other people.

The third stage is aggregation of contribution. During this stage each contribution is unified into a final "product" that represents the entire pool of contribution. For instance, the aggregation for Wikipedia articles generally happens right away, barring articles that need special attention. For Stack Overflow, users with enough reputation points can upvote and downvote other users' questions and answers. The users who posted the question can also mark and reply to the answer.

The fourth stage is the Remuneration of contributors. During this stage, the contributors are credited for their contributions in various ways, such as having their names credited as contributors, monetary compensation, or simply no credits (entirely voluntary works). For Wikipedia, users who contributed to any articles can be seen in the edit history. For Stack Overflow, the users who contributed good questions and answers are awarded with reputation points, which grant them privileges such as the aforementioned edit access, upvoting, and downvoting questions and answers.

1.2 Wisdom of Crowds

Of course, not all crowds are guaranteed to be wise and rational, and not all crowdsourced objects (tasks, articles, resources, etc.) are as good as expert-created ones. According to Surowiecki [51], there are four major conditions that ensure effectiveness of crowdsourcing. First, the crowd needs to be diverse, each member of which adding their knowledge and information. Second, the crowd needs to be independent of each other; each crowd input should not affect others' input. Third, the crowd needs to be decentralized; each crowd input is created based on their unique or specialized knowledge. Fourth, all crowd inputs must be properly aggregated; each input should be properly taken into account. Aggregation method is the key to a successful crowdsourcing system [12]. Given the widespread usage of the internet around the world, these four conditions are now easier than ever to satisfy. As a result, several organizations and platforms around the worlds have shifted toward crowdsourcing, and many have shown to be successful.

Crowdsourcing, however, is not without disadvantages. There are three major challenges with crowdsourcing [38][14]. First, the crowds may not perform the task. Contrary to traditionally hired experts, the crowds generally are not motivated or incentivized to perform tasks. In addition, the crowds are concerned about intellectual properties. Free-loaders, people who are only "consumers" of tasks and never contribute, may cause systems to fail to sustain due to lack of participation. Second, while aggregated crowdsourced contents are almost always as good as, if not better than, expert-created contents, most individual crowdsourced contents are not. An individual crowd member often operates with their own limited information and biases and, as a result, is unable to provide quality contents. In addition, many may choose to purposefully spread false information and attempt to sabotage, especially in an anonymous environment. Third, even when every individual crowd member contributes with their best effort and the four conditions above are satisfied, the aggregated contents may have done nothing but reinforcing what the dominant voices in the crowds know or believe, this effect is also called an ”echo chamber.”

Chapter 2

Examples of Crowdsourcing in Educational Research and Applications

There are several crowdsourcing works in educational research and applications, with varying degree of implementation from prototypes to deployed products. In this section, I will list some well-known problems in educational research and how crowdsourcing can be used as a solution for such problems.

2.1 Crowdsourcing Assessment and Grading

Feedback and assessments are generally deemed beneficial to learning. For open-ended responses or essay questions, it's hard for instructors to provide feedback and assessments to all learners in MOOC environments for many reasons such as sheer numbers of learners, who may start at different days and learn at different rates. There are several solutions that are widely used:

1. self-assessment: self-assessment is very easy to implement, but it is often unreliable as learners tend to rate themselves too highly. There are works that incorporate mechanism to discourage students from over-rate themselves such

as [25]

2. peer-assessment: peer-assessment is when the grading is redistributed to other learners in the same courses. Learners are often organized in pairs or small groups for peer assessments.

3. paid freelance teaching assistant (e.g. Cloud Teaching Assistant System in Iversity). In [52], they found that the grading of cloud teaching assistants is highly correlated to that of peers (Pearson's correlation 0.76). While cloud teaching assistant is a good substitute for peer grading, neither of them simulated instructor grading (Pearson's correlation of 0.36 and 0.39, respectively).

4. artificial intelligence grading (e.g. AI Grading in edX [13], [30]. If the model is trained such that it can predict instructors' grading, AI grading would be able to provide the feedback to the learner essays instantly, which is a huge benefit over other methods. However, the model is also very susceptible to data that are not represented in the training set. It could also be gamed by clever learners. AI grading also can't be deployed on a new course or a new topic because the model has to be trained with graded submissions.

2.1.1 Peer assessment

In peer assessment, after learners finished their questions, they will be asked to grade a subset of responses from other learners who worked on the same question. Peer assessment not only provide feedback and assessment to learners, but the act of assessing other learners' response is also believed to improve learners' "sense of ownership and autonomy, increased motivation, enhanced social presence, and the development of higher-order thinking and meta-cognition skills." [27]. Many applications also include self-assessment as an addition step to peer assessment as well [9].

There are two main concerns of peer-assessment validity and reliability. Validity is usually defined as the correlation between peer-assessment grading and instructors grading. Reliability is usually defined as the correlation between the grading of multiple different peer graders [27, 16]. There are studies that suggested that peer grading and instructing grading are closely correlated, such as[16], and otherwise such as [52]. Peer-assessment is also often deemed unreliable by learners and instructors, especially in MOOCs where the background knowledge of learners vary greatly citecapuano2016improving.

2.1.2 Improving Reliability and Validity

Calibration is the most common method to improve reliability and validity of peer-assessment. In calibrated peer assessment, learners go through an additional step (”calibration”) right before they’re asked to grade other learners’ responses. In this calibration steps, learners are asked to grade a few benchmark examples using specified rubric which were previously graded by instructors. In many learning systems, learners are given feedback and asked to grade more until they reach satisfactory accuracy (e.g. edX) [9, 3].

2.2 Crowdsourcing additional instruction from peer

Peer Instruction is one of the most widely accepted active learning pedagogical strategies. A lesson with peer instruction follows steps like

1. the teacher presents a question to the class for students to answer.

2. the teacher then asks students to discuss their answer with their neighbors, and convince each other if they disagree.

3. the teacher asks the original question to the class again, and the answer could be the same or different, and maybe for different justification.

The process of peer instruction is hard to emulate in online learning environments, especially in many platforms where students work asynchronously.

Peer Instruction could be considered crowdsourcing rationales from students, not only the correct ones but also the incorrect answers and correct answers with incorrect explanations as well. Erroneous examples have been shown to improve student learning [1, 20].

2.2.1 DALITE

[6] created a system called Distributed Active Learning Integrated Technology Environment (DALITE) that supported for peer instruction during the online learning. DALITE focuses on multiple choice questions, as it is easier to solicit and group rationales from students.

DALITE follows the 3 steps similar to peer instruction above.

1. DALITE displays a multiple-choice question, then prompt students to write a few sentences explaining their answers (rationales).

2. DALITE then displays 2 sets of 4 rationales, one set for their chosen answer, and one set for another choice. These rationales are crowd-sourced from all previous students who answered this question before, so it can be done asynchronously. DALITE asks students to reflect on their thinking using the provided rationales and whether they change their mind. DALITE also asks students to vote on which rationales they like the best.

3. DALITE shows the question with the answer students chose in step 1 and step 2, along with corresponding rationales.

This process allows DALITE to roughly understand each peer instruction without requiring complex language model. In fact, this process itself can also be used on problems of different languages or new problems right away without pre-training unlike model-based approaches (though peer instructions themselves still can't be used across languages).

2.2.2 PeerASSIST

PeerASSIST was a feature inside ASSISTments that allowed students to receive additional instruction from their peers [48]. Unlike DALITE, PeerASSIST wasn't limited to multiple choice questions, but it required ”show your work” to be enabled. ”Show your work” was another feature inside ASSISTments that teachers can enable for particular assignments. When this feature was enabled, when students submit their answer, they would have to also provide ”their work” in a form of rich text, which may include formatting, images, and videos. If the student answered incorrectly, they could submit their new work when they re-submit an answer.

When a student answered a problem correctly on their first attempt, PeerASSIST treated the corresponding student’s work as a possible peer instruction (called peer explanation in PeerASSIST context). Such peer explanations were then displayed to other students who struggled in the same problem, defined by having exhausted all partial credits (by default, partial credits were exhausted after 3 incorrect attempts).

Such approach would allow PeerASSIST to gather a large number of peer explanations. However, there were three problems. The first problem was the fact that students’ works, while they were worked examples, were written in lesser details since they aim only to show to their teachers that they had correct knowledge and concepts. The second problem was the fact that peer explanations could still be erroneous even though they had answered the problems correctly. The errors could range from minor issues, such as incorrect terms, to major issues, such as using incorrect formula (that incidentally gave the same answers). In addition, students’ works were unsuitable to be distributed outside of the teachers’ own classes due to privacy, which is the third problem. To my knowledge, PeerASSIST did not attempt to solve the last problem and left it as a limitation.

PeerASSIST pruned such undesirable peer explanations (problem 1 and 2) by using two methods. The first method was using bandit algorithm. After students received peer explanations given out by PeerASSIST, PeerASSIST looked at student performance after said problems (such as how many attempts they made and correctness) compared to student prior performance (such as their average attempt count and percent correct).This information was then used to calculated the score for each peer explanation, allowing PeerASSIST to be smarter about which peer explanation to be given out next time.

A second, more teacher-driven method was to allow teachers to designate some of their students as starred students. This is analogous to a practice that some teachers gave tokens (generally star-shaped) to students who have made good and/or consistent progress in their study. When starred student feature is enabled in PeerASSIST, PeerASSIST would only choose and display peer explanations for starred students. Teachers could then tell their starred students that their works would be displayed to their struggling peer, and instruct them to write their works with more care and clarity.

2.3 Crowdsourcing Student Supports (e.g. Hints)

During in-class practices, students could ask their teachers for help when they struggle on their assignments or practices. In an online learning environment, this is replaced by computer-provided student supports, such as hint messages, breaking the problems into smaller steps, and full solutions of the problem /citerazzaq2009tutor. Several studies shown that such student supports increased learning outcomes [33, 41, 4, 50, 5, 21].

In the field of educational research, there had been several works that attempted to scale up such student supports, problem-specific and otherwise, such as using model-based hint-generators for problem solving problems [28] and using crowd sourcing [53].

2.3.1 A Crowdsourcing Approach To Collecting Tutorial Videos – Toward Personalized Learning-at-Scale

In this work by Whitehill and Seltzer, they explored how to crowd source video explanations on how to solve logarithmic problems from ”teachers” and investigated their effectiveness. Specifically, they gathered instruction videos from Amazon Mechanical Turk workers (”teachers”). They obtained a total of 399 videos. Of 145 videos they sampled and investigated in the paper, 117 videos were found to be mathematically correct. They further randomly sampled down to 40 videos and used 200 Mechanical Turk workers ("learners") to find the best videos. They gave the learners pretest on logarithm, then the video as a lesson, and posttest on logarithm.

They conducted the same experiment comparing 4 best videos from the previous experiment to Khan Academy video on logarithm (as the control) using 250 MTurk learners (participants were uniformly randomly assigned to one of the 5 videos). They found that the best crowd sourced video was comparable to Khan Academy in term of learning gain. They also noticed that, in addition to teaching experience, Khan Academy video was substantially longer than their crowd sourced videos, and their crowd-sourced videos were more closely aligned with their pre- and post-test.

Chapter 3

Crowdsourcing Randomized Controlled Trials using Experiment Templates

3.1 Introduction

ASSISTments is a free online educational tool for teachers and students. Inside ASSISTments, teachers can create their own problems and problem sets, assigning problem sets, and manage grade books. At the same time, ASSISTments provides many features that help reduce teacher workload such as automated grading and providing a library of problem sets, which is maintained by learning scientists.

ASSISTments always strives to make its contents better. To do so, ASSISTments always runs randomized controlled trials (RCTs) not only to improve its contents but also the understanding of human learning in the field of learning science. While many RCTs are proposed by the learning scientists and graduate students working on ASSISTments and their collaborators, independent researchers can also propose studies through the ASSISTments Test Bed [34]. Most RCTs run inside ASSISTments are experiments embedded in skill builders, which are mastery-based problem sets. It can be said that the ASSISTments Test Bed is a tool for crowdsourcing RCTs from

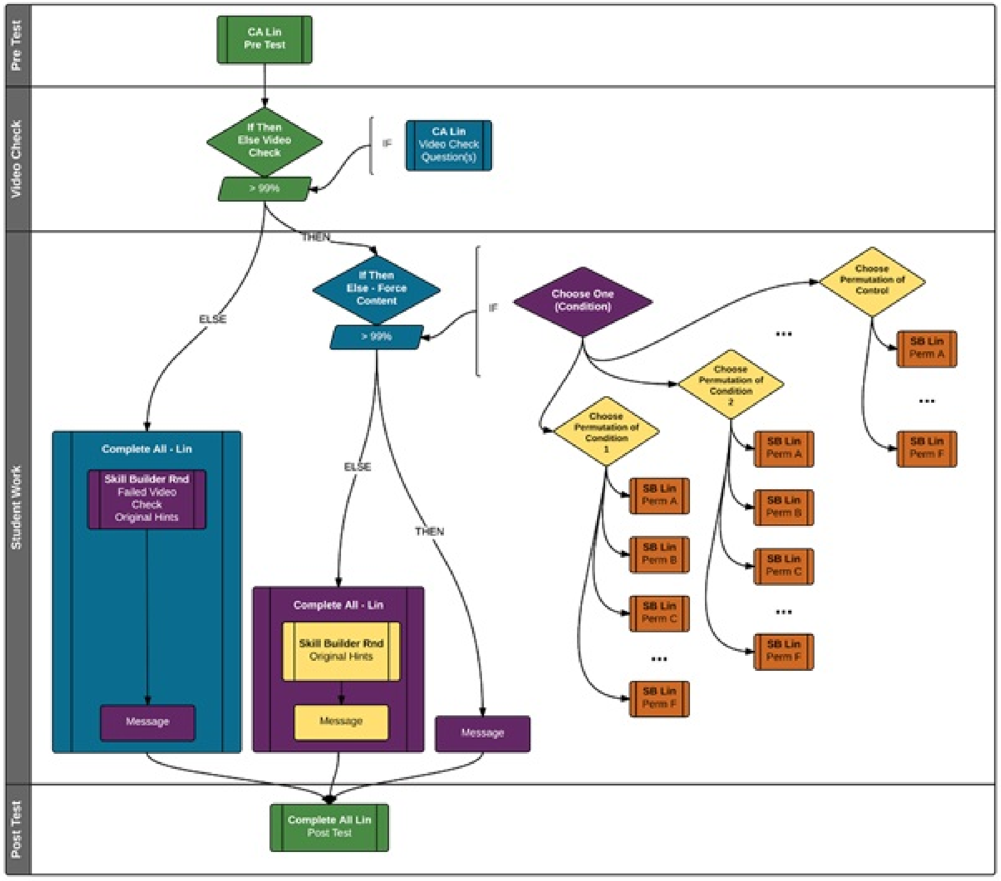

Pre Test
Video Check
Student Work
Post Test
CA Lin Pre Test
If Then Else Video Check
IF
CA Lin Video Check Question(s)
> 99%
THEN
ELSE
If Then Else - Force Content
IF
> 99%
Choose One (Condition)
Choose Permutation of Control
Choose Permutation of Condition 2
Choose Permutation of Condition 1
SB Lin Perm A
SB Lin Perm B
SB Lin Perm C
SB Lin Perm F
...
Complete All - Lin
Skill Builder Rnd Failed Video Check Original Hints
Message
ELSE
THEN
Complete All - Lin
Skill Builder Rnd Original Hints
Message
Message
Complete All Lin Post Test

Figure 3-1: An experimental design of a video vs text RCT with pre-test, post-test, and video check.

researchers.

There are many limitations and inconvenience of the ASSISTments Test Bed. For example, embedding an experimental structure into a problem set is a complicated and error-prone task, even for experts who had run several RCTs inside ASSISTments. In addition, researchers, especially those who are not familiar with ASSISTments problem set structures, such as in Figure 3-1, may misinterpret the dataset from their experiments due to the complexity of the experimental structure in the RCT problem set.

In order to reduce the complexity in the process of creating and analysing RCTs inside ASSISTments, the ASSISTments team came up with several RCT designs (”patterns”) for skill builders ([2]). Each pattern is a commonly used experimental design (such as video vs text, with optional pre-test and post-test).

3.2 ASSISTments Build-from-Pattern

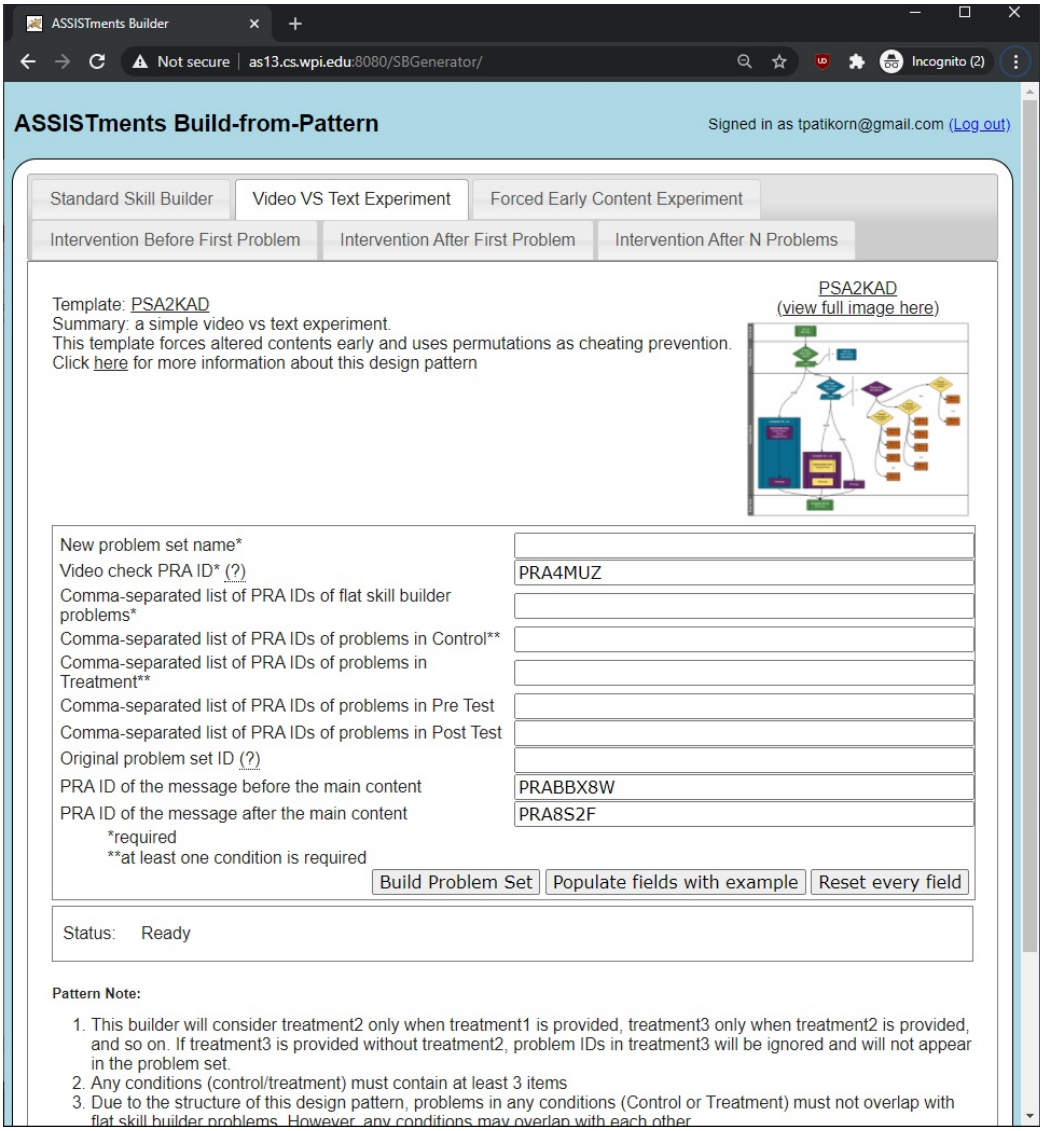
While patterns are helpful to researchers as blueprints for experiments, it does not necessarily mean that researchers would be able to translate that into the a functioning experiment inside ASSISTments, especially without the help of research scientists working for ASSISTments.

In 2016, I created a web-based tool that automate the progress of constructing the experiment based on such experiment pattern, shown in Figure 3-2. The tool is form-based; researchers simply have to fill in which problems they want to show in each section of the RCT, from pre-test, control condition, experimental condition, to post-test. When the researchers click "Build Problem Set," the tool will check for the validity of the given input and, if the given information is sufficient, build a problem set with a specified pattern and input problems.

3.3 Result

Unfortunately, the ASSISTments Build-from-Pattern has not been used much for various reasons.

First, there had been low number of RCTs created from when the tool was created, especially by researchers outside of ASSISTments. Second, while patterns capture main ideas of experimental designs, there are almost always little details that the patterns, when the tool was created, fail to capture. For instance, in a pattern for "Intervention After First Problem," the intervention is supposed to show up after the student in the experimental condition answers their first problem incorrectly. The pattern, however, did not capture the case where researchers want to have the intervention shown when the student makes their first incorrect problem, regardless of whether that problem is the first problem the student encounters or not. Third, the tool lacks the ability to save input as drafts.


ASSISTments Builder
Not secure | as13.cs.wpi.edu:8080/SBGenerator/
Incognito (2)
Standard Skill Builder
Video VS Text Experiment
Forced Early Content Experiment
Intervention Before First Problem
Intervention After First Problem
Intervention After N Problems
PSA2KAD
(view full image here)

ASSISTments Build-from-Pattern

Signed in as tpatikorn@gmail.com (Log out)

Template: PSA2KAD
Summary: a simple video vs text experiment.
This template forces altered contents early and uses permutations as cheating prevention.
Click here for more information about this design pattern

New problem set name*	
Video check PRA ID* (?)	PRA4MUZ
Comma-separated list of PRA IDs of flat skill builder problems*	
Comma-separated list of PRA IDs of problems in Control**	
Comma-separated list of PRA IDs of problems in Treatment**	
Comma-separated list of PRA IDs of problems in Pre Test	
Comma-separated list of PRA IDs of problems in Post Test	
Original problem set ID (?)	
PRA ID of the message before the main content	PRABBX8W
PRA ID of the message after the main content	PRA8S2F

*required
**at least one condition is required

Status: Ready

Pattern Note:

1. This builder will consider treatment2 only when treatment1 is provided, treatment3 only when treatment2 is provided, and so on. If treatment3 is provided without treatment2, problem IDs in treatment3 will be ignored and will not appear in the problem set.
2. Any conditions (control/treatment) must contain at least 3 items
3. Due to the structure of this design pattern, problems in any conditions (Control or Treatment) must not overlap with flat skill builder problems. However, any conditions may overlap with each other

Figure 3-2: ASSISTments Build-from-Pattern: a form-based experiment builder based on design patterns described in the ASSISTments Test Bed

To link back to Surowiecki's Wisdom of Crowds, it could be aside that our crowds are diverse and decentralized, causing the experiments they want to run to differ. Our patterns were too inflexible and thus were unable to capture the variance in the experimental designs.

Chapter 4

Crowdsourcing Data Mining Solutions through Competition: ASSISTments Longitudinal Data Mining Competition 2017

4.1 Introduction

During the 10th International Conference on Educational Data Mining in Wuhan, China, the ASSISTments Longitudinal Data Mining Competition was announced by the Big Data for Education Spoke of the Big Data Northeast Innovation Hub, a research hub funded by the U.S. National Science Foundation. This competition used a longitudinal dataset collected on students using ASSISTments, a free online tutoring platform, in 2004 - 2006. The ASSISTments team tracked those students to see who graduated from high schools, who went on to college, what their majors were, and finally if they chose a career in STEM (Science, Technology Engineering and Math) for their first job, post-college. Several papers have shown that behavior in ASSISTments in middle school can predict high school and college outcomes [32] [39] [45]. The task given to the participants in this competition was to use deidentified click-stream data to try to predict the whether the student pursued a career in STEM or not. This data was provided to participants to analyze before it was used by the research team themselves, an unusual step that enabled participants in the competition to gain first access to a cutting-edge research data set.

In recent years, there has been increasing interest by school districts and state education agencies in predicting student success and dropout [7] [23]. These detectors are used to give early warnings to teachers, guidance counselors, and school leaders when students show signs that they are losing interest or experiencing difficulties. These detectors support teachers making targeted interventions to take necessary actions to help students before it's too late. However, there has thus far been relatively less work to drive K-12 early warning based on students' risk of dropping out the STEM pipeline. This is particularly problematic, given the current economic context. While there is increasing demand for STEM workers, substantial numbers of students lose interest in STEM subjects and fields or are insufficiently prepared to participate in these careers [46]. Developing automated detection of STEM career participation may help us to identify students who could benefit from an intervention to help to support their interest and readiness for STEM [42].

4.2 ASSISTments Longitudinal Data Mining Competition 2017

The competition ran from June 27, 2017 to December 3, 2017. Registration for the competition and the dataset were entirely free, in line with the goals of promoting 1) STEM education, 2) educational data mining, and 3) open science. The primary condition of accessing the dataset was to not take any action to deanonymize the dataset. Even though the competition has already been concluded, we still welcome interested researchers to sign up for the competition dataset.

4.2.1 Dataset

The dataset in this competition was the ASSISTments clickstream dataset collected during 2004 - 2006. This dataset contained actions middle-school students took while working on their mathematics assignments. In addition to raw recorded actions, participants were also provided with several distilled measures, for instance, measures of the student's affective state and disengaged behaviors (bored, concentrating, confused, frustrated, off-task, and gaming). These measures were obtained by collecting student affect observations in real classroom and then using machine learning techniques to train models that replicated those judgments within a clickstream dataset [35]. The detectors were validated to ensure that they applied effectively to unseen students from urban, rural, and suburban settings [31]. The dataset contains 78 clickstream data predictor variables and the target variable "isSTEM": whether the student's career of choice was in the STEM fields or not, defined using the NSF guidelines for STEM careers. There are 942,816 action-level data rows collected from 1,709 students in total. For the competition, the dataset was split into 3 sets: the training set, the validation set, and the test set.

Training Set

The training set contained the majority of the students from the full dataset. For each student in this dataset, both the students' action-level ASSISTments usage data and their "isSTEM" variable were available. Participants, as well as any researchers who are interestedin STEM education, could make full use of this dataset, using any state-of-the-art data mining technique they chose to find the relationships between the student actions and their career choice (as long as it does not violate the terms of use).

During the data collection, there were many students for whom we collected ASSISTments usage data, but we were unable to retrieve their career information. Specifically, we know the isSTEM for only 591 students out of 1,709 students. We decided to include the ASSISTments usage data of these students in the training set since there are many co-training machine learning approaches that could train a model by using unlabeled data along with labeled data. The training set contains 514 labeled students and 1,118 unlabeled students.

Validation Set

The validation set was mainly used for the public leaderboard. This leaderboard let participants know how well they were doing compared to other participants. All clickstream data from students in the validation set were made available to participants. Participants, however, were unable to directly access the ”isSTEM” variable for the students in the validation set. When ready, participants could submit their prediction for the validation set's isSTEM students. The system would then evaluate the predictions, inform participant of their scores, and then update the participant's best scores on the leaderboard. The evaluation scheme will be further discussed in the later section.

Test Set

The only purpose of the test set was to be used to determine the winner of the competition. Like the validation set, participants could only access the clickstream data of students in this set and not their isSTEM. The difference between the validation and the test set was that the test set was not used to calculate the leaderboard scores; the results were not visible until after the competition was complete. The reason we chose to separate the test set from the validation set was to make sure that the winners of the competition were not simply participants who overfit using the leaderboard, but who genuinely could predict entirely unseen data.

4.2.2 Evaluation

For the evaluation of models, participants were required to submit their predictions for students in both the validation set and the test set. Participants, however, were not informed as to which students were in which set. Once a day at noon EST, new submissions were evaluated on the validation set. While participants could submit as many predictions as they wanted, only the participant's latest submission was evaluated, to discourage them from overfitting to the leaderboard. The system then updated each participant's personal submission log with their latest submission's scores as well as the public leaderboard, where each participant's best scores were shown compared to other participants' best scores.

Evaluation Criteria

Both the leaderboard scores and the final scores were calculated by using a linear combination of the area under the ROC curve (AUC) and the root mean squared error (RMSE). Since isSTEM was observed and collected as binary values, AUC was initially chosen as the evaluation criterion. AUC captures the model's ability to differentiate students in the two categories from each other, based on the relative confidence in the predictions. It is most suitable when the variable being predicted is binary and the predictions are numerical. However, after testing, we found that AUC, or any single metric, could be easily overfit to, especially given the small sample size.

Thus, we selected a second evaluation criterion: RMSE. While RMSE is designed for comparing two numbers, it provides an assessment that rewards models that are more certain when they are correct and punishes models that are uncertain with high confidence. It also maps to a context of use where the model provides different recommendations when it is uncertain than when it is highly confident.

For the sake of the competition, we decided to aggregate the two metrics, AUC and RMSE, into one score so that we could determine the winners. Since AUC ranges from 0 (reverse ranking) to 1 (perfect ranking) and RMSE, in this case, ranges from 0 (perfect predictions) to 1 (total opposite predictions), we define Aggregated Score as a linear combination of the two metrics, with one metric inverted: $AggregatedScore = AUC + (1 - RMSE)$

4.2.3 Different Population from Training to Validation and Test Sets

In October 2017, we discovered that the distribution of isSTEM within the training set was not the same as that of validation and test set. Specifically, the ratio of isSTEM = true and isSTEM = false of the validation set and test set were the same, but that ratio of the training set was more than double that of the validation set and test set. We investigated the issue and decided to keep the three sets as they were and announced this information to all participants. The reasons we decided to keep the data sets unchanged were 1) it is not uncommon for models to be applied to a context with different distribution and/or population from the training set. The difference between the sets, while they were not intended, did emulate this possible real application issue. 2) the isSTEM ratio of the validation set and the test set were the same, meaning participants could use the result from the validation set to adjust for the discrepancies between the training and the validation set, which would be reflected in the test set, since the isSTEM distribution of the validation and test sets were the same.

4.3 Conclusion of the Competition

The competition was concluded on December 3rd, 2017. At the conclusion of the competition, 202 participants had signed up for the competition, 74 of whom submitted predictions at least once.

4.3.1 Data Request Over Time

Most of the requests for the dataset were from August 2017 to November 2017. Since one of our main goals is to promote research in this area, we were glad to see that requests for the dataset continued even after the competition ended in December.

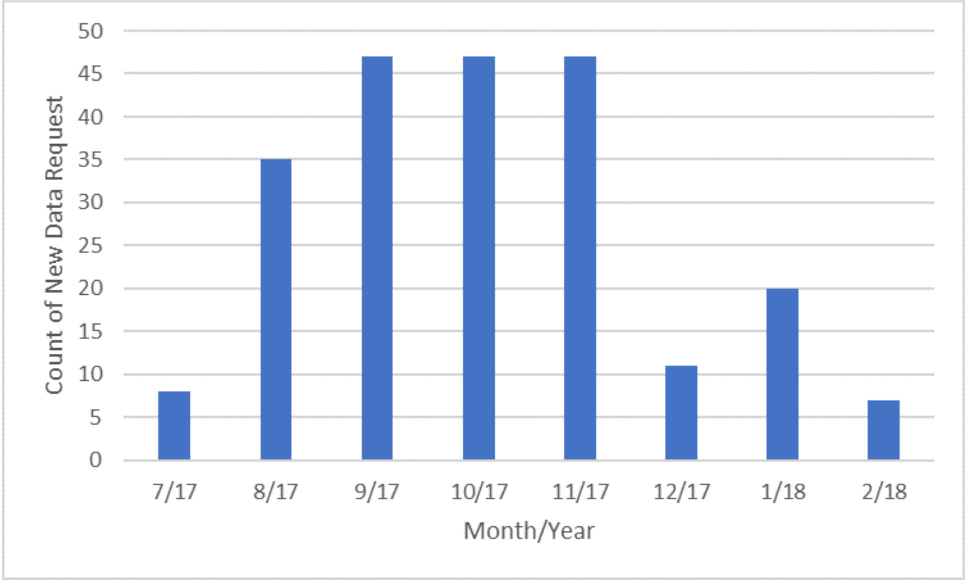


Figure 4-1: the number of new unique emails that signed up for the competition dataset in each month from July, 2017 to February 2018.

4.3.2 Submissions Over Time

At the first glance, the number of submissions peaked during November 2017, which was the last full month before the competition concluded. However, since the competition concluded on December 3rd, 2017, December 2017 was the month with the most submissions per day of 19.33, more than double the rate in November 2017 (9.19 submissions per day). Among all participants who submitted predictions at least once, about two-third of them submitted more than once, and only about one-sixth submitted more than ten times. Only 8 participants submitted more than 20 times.

4.3.3 Submissions Scores Over Time

Overall, the quality of submitted predictions averaged across all participants appeared to increase slightly over the months as shown in Figure 4-4. While the average scores seemed to plateau after October, it is important to note that there were many participants who joined later in the competition. Their scores were averaged together with other participants who had already worked on the competition. We further

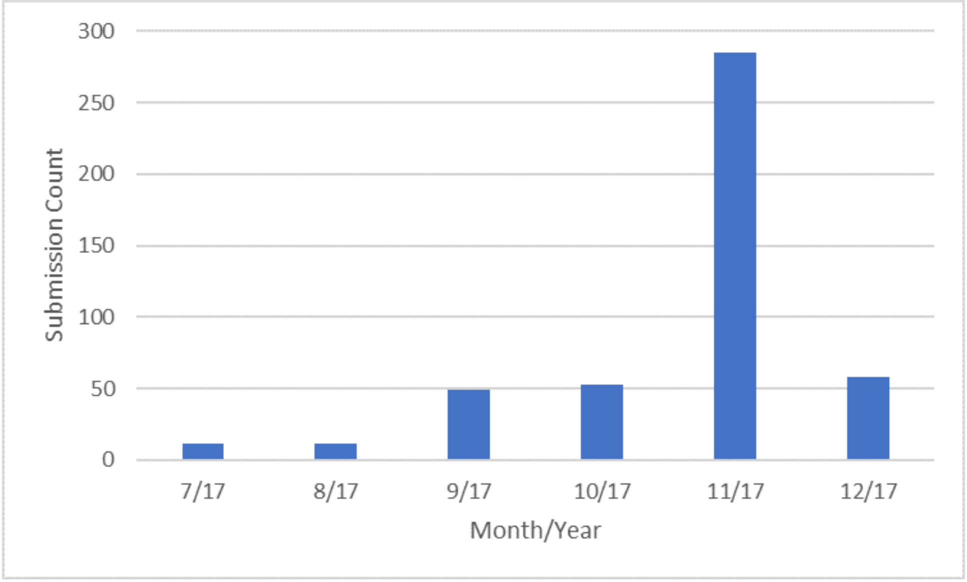


Figure 4-2: the number of submissions evaluated by the system in each month from July, 2017 to December 2017.

investigated by looking at the aggregated score of the 1st, 2nd, 3rd, etc. submissions averaged across all participants, which is shown in Figure 5. A similar increasing trend to Figure 4 can also be observed in Figure 4-5. It is important to note that there were only 8 participants who submitted more than 20 times, which could be one of the reasons why the graph fluctuates a lot when x > 20.

4.3.4 Winners

The three winners were announced during the NorthEast Big Data Spoke Meeting at MIT on February 16th 2018. The first place winning team of Chun Kit Yeung, Kai Yang, and Dit-yan Yeung is from the Hong Kong University of Science and Technology, who participated in the workshop. The second place winner was Makhlouf Jihed from Japan's Kyushu University, who also participated in the workshop. The third place honors went to the University of Michigan Data Science Team, a group that regularly competes in data competitions like this one.

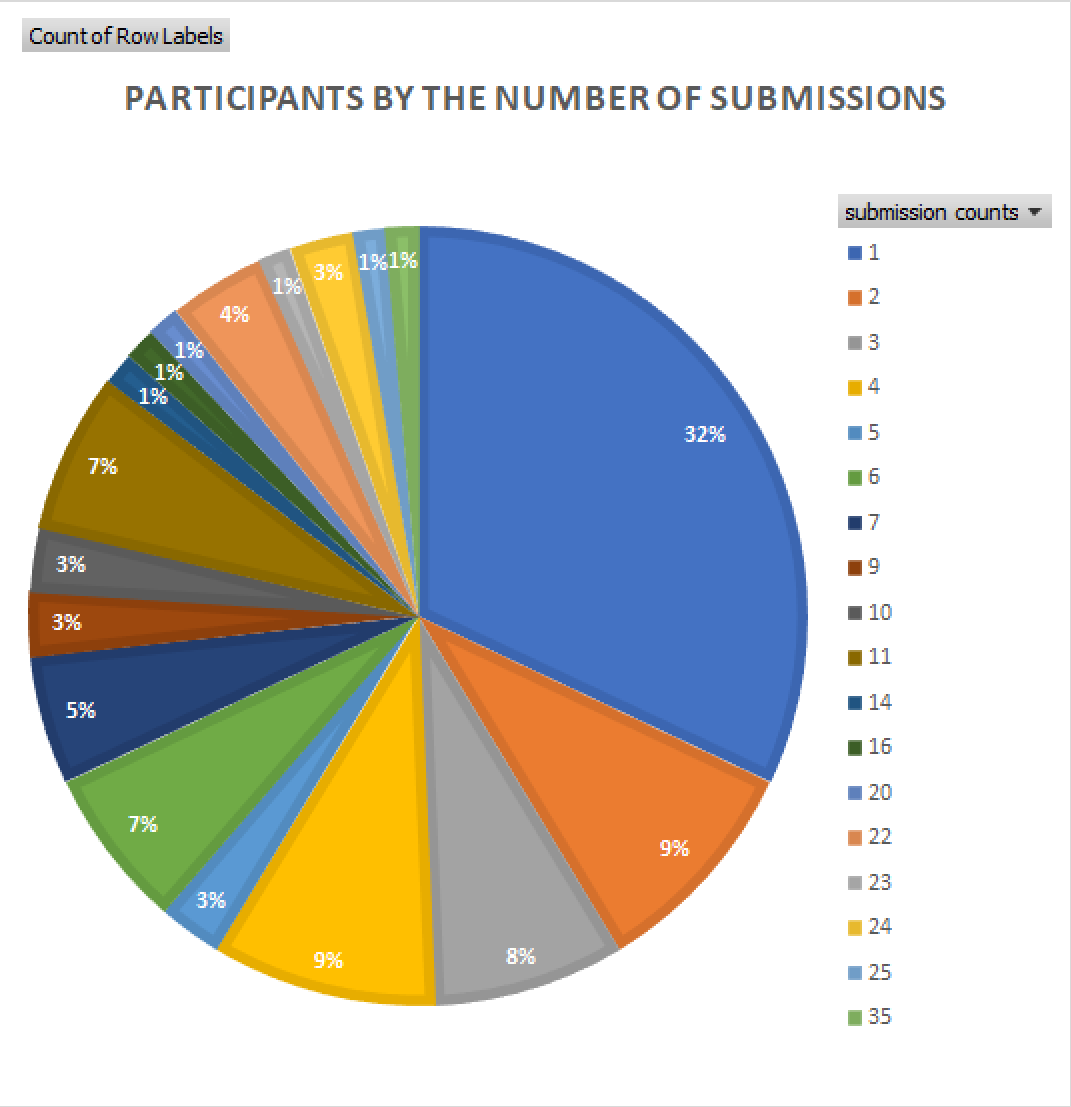


Figure 4-3: the percentage of participants by the number of submissions they made during the competition.

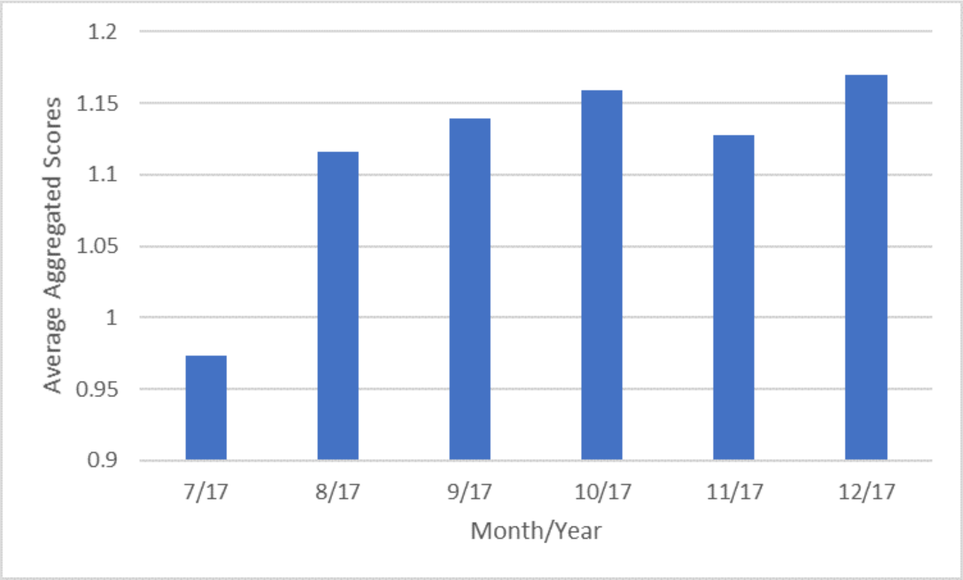


Figure 4-4: the aggregated scores averaged across all participant predictions submitted and evaluated in each month from July, 2017 to December 2017.

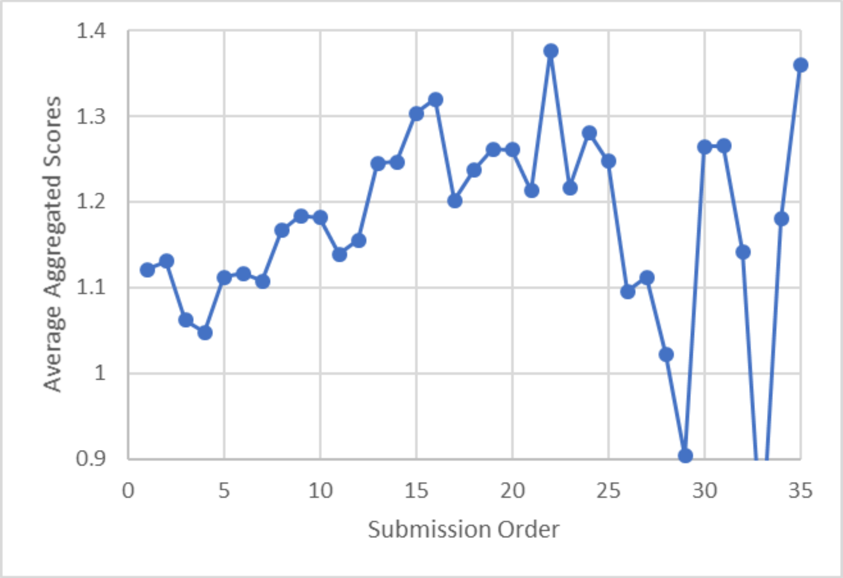


Figure 4-5: the aggregated scores by the submission order of each participant, averaged across participants from July, 2017 to December 2017. For example, the average aggregated scores of everyone's second submission is the data point at x = 2.

Chapter 5

Crowdsourcing Data Mining Solutions through Competition: Nation's Report Card Data Mining Competition 2019

5.1 Introduction

During the 20th International Conference on Artificial Intelligence in Education in Chicago, Illinois, the Nation's Report Card Data Mining Competition was announced by the Big Data for Education Spoke of the Big Data Northeast Innovation Hub, a research hub funded by the U.S. National Science Foundation. The goal of this competition was to engage leading researchers and promising doctoral students in a Grand Challenge that pushes the field of educational data mining forward, develops metrics for measuring students' test taking activities, and help develop and test evaluation methods for educational analysis. Competition participants were invited to assess data produced by students early in a test to predict students' future activities later in the test. Thus, competition participants would try to understand effective and ineffective test-taking behaviors, and to determine how quickly these behaviors can be detected.

This competition was designed to improve the scientific understanding of student test-taking strategies. The results of this competition show that as early as two minutes into the test, the best of these algorithms could predict with 65% accuracy whether or not the data was from a student student who was not as motivated in the second half of the test.

Professor Neil Heffernan, director of the PhD program in Learning Sciences and Technology at WPI and one of the organizers, "The Nation's Report Card's mission is to show the trend line of our nation's progress in developing student knowledge. This competition is one step in helping to improve our understanding of the NAEP, as there is a concern that students might not be taking the NAEP test as seriously as they used to. For instance, we could use this data to identify a student who is potentially not motivated throughout the test, and between sections, invite the student's teacher to offer encouragement. It's too early to know how NAEP should use these algorithms, but this competition could be an important step in developing appropriate interventions"

This competition concluded on December 15, 2019 at 11:59 p.m. EST. On March 11, 2020, we announced three winners and two honorary mentions based on their aggregated scores on the test set.

5.2 Dataset

The competition used dataset provided by Educational Testing Service, with permission from The Nation's Report Card, also known as the National Assessment of Educational Progress (NAEP). The NAEP is the only assessment that measures U.S. student knowledge nationwide across academic subjects. The NAEP has collected data since 1969 and measures student success in urban, suburban and rural areas.

This dataset was a deidentified compilation of actions students made during testing in the 2016-2017 academic year. The students worked on "blocks" of test math problems, referred to as Blocks A and B. Each block contains a set number of problems and each student had a 30 minute time limit to complete the problems in each block. Once the 30 minutes are completed, students are automatically dismissed from the block, regardless of how many problems they have completed. Please view several sample questions from the 8th grade curriculum.

5.2.1 Target Variable

The Target Variable was a binary indicator of whether or not the student spent their time in Block B efficiently. Specifically, we defined efficient usage of time as 1) being able to complete all problems in Block B, and 2) being able to allocate a reasonable amount of time to solve each problem.

We defined a "reasonable amount of time" as the minimum possible time needed to solve each problem. This threshold is very hard to define. For the sake of this competition, we chose the threshold based on the distribution of the total amount of time students spent on each problem in the dataset. Specifically, for each problem in Block B, we ranked the total amount of time each student took to complete each problem, and used the 5th percentile as the cut-off for the "reasonable amount of time."

5.2.2 Training Set and Hidden Set

We separated the dataset by students into subsets: the training set and the hidden set. The training set is provided to allow participants to build models to predict whether students in the hidden set spent time efficiently in Block B, using only (some of) their data from Block A.

1. Training Set: For each student in the training set, we provide all 30 minutes of their logged actions in Block A, as well as whether they spent their time efficiently in Block B or not (target variable)

2. Hidden Set: The target variable is not provided for any students in the hidden set. The hidden set consists of 3 components of equal portion. For each component, we provide different amounts of information from Block A. Specifically:

(a) For the first component, we provide all 30 minutes of logged actions similarly to the training set

(b) For the second component, we only provide the first 20 minutes of logged actions (the last 10 minutes of logged actions were omitted from the dataset).

(c) For the third component, we only provide the first 10 minutes of logged actions (the last 20 minutes of logged actions were omitted from the dataset).

We then created a leaderboard set and a final test set, of equal size, drawn equally from the three components. The leaderboard set is used to provide participants with feedback on how their models perform in comparison with other participants, when applied to half of the hidden set. The final test set is the subset that will be used to evaluate participants' prediction at the end of the competition. In creating the subsets and the leaderboard and test sets, as well as the three components, we maintain the original distribution of the target variable in all cases.

5.3 Result

The Nation's Report Card 2019 Data Mining Competition had 89 individual and team participants in the competition, totaling 723 submissions. Researchers and students from 11 countries and 24 U.S. states participated in the competition. Some of the research teams were made up entirely of undergraduates. The organizers are pleased that this competition inspired undergraduates to care about educational data and become interested in its use in research.

5.3.1 Winners

Winners were judged based on the final score of their submission using the evaluation criteria specified in our competition website.

The first place winner was Nathan Levin from Teachers College, Columbia University in New York City. He constructed and refined features based on student click data and the time students spent working on problems. He then applied XGBoost Regressor on the final feature set.

The second place winners were Nirmal Patel, Aditya Sharma, and Tirth Shah from Playpower Labs. They constructed a large number of features using the results of their previous research, many of which were inspired by Process Mining and Curriculum Pacing. They then applied Genetic Algorithm-based feature selection and modeling. The predictions from multiple models were then assembled together to create a single final prediction.

The third place winner was Assistant Professor Nigel Bosch from the iSchool at the University of Illinois Urbana-Champaign. He constructed a large number of features ($>$ 4,000) using both domain knowledge and automatic feature engineering methods, specifically TSFRESH and FeatureTools.

Participants of the top submissions will receive an invitation to submit their work and findings to a special issue of the Journal of Educational Data Mining. This should help to further improve the field's understanding of this important work.

5.3.2 Honorary Mentions

Among all of the participating teams, two additional teams showed outstanding efforts and achieved impressive results in both the leaderboard and the final test set: KLETech B Division from KLE Technological University (Huballi, India) and LTWZ from the Columbia University (New York City) and the University of Arizona (Tucson).

KLETech B Division treated the hidden dataset as three different tasks and developed a model for each task based on the different amounts of information provided (e.g., only the first 10 minutes, only the first 20 minutes, and all 30 minutes of log data). LTWZ developed their model using features based on student test-taking behaviors, such as the frequency of how often each student checks the test timer.

Chapter 6

Effectiveness of Crowd-Sourcing On-Demand Assistance from Teachers in Online Learning Platforms

6.1 Introduction

In recent years, the usage of digital learning media in K-12 classroom has grown exponentially. From the teacher and student perspectives, online learning platforms allow for new ways of learning that may otherwise be hard or impossible to do such as individualized mastery learning [10, 19]. Possibly one of the most important features of these systems is the ability to assist students as students work on their assignments. The most common type of assistants is answer feedback where the students know right away if their submitted answers are correct or not. In this work, we are interested in on-demand assistance. This type of assistance, sometimes called "tutoring," provides students with an option to request additional resources that would help them solve the problems, such as hint messages or complete explanations. The ability to provide students with additional guidance while they are outside of classrooms is especially valuable for homework assignments and distance learning such as during the COVID-19 pandemic. Since on-demand assistance is problem-specific, creating and maintaining on-demand assistance is hard and time-consuming. The cost of on-demand assistance scales with the number of problems in the system. For instance, out of 132,738 distinct problems assigned by all teachers inside ASSISTments in the 2017-2018 academic year, only 38,194 of them had on-demand assistance.

During a large-scale evaluation of the ASSISTments online learning platform [44], the intervention consisted of three components, supporting their 1) textbook work 2) skill builders (adaptive skill practice), and 3) teacher-created contents. First, related to the textbook work, we allowed each teacher to keep using their current textbook; we did the data entry to put the answers to the textbooks' questions into ASSISTments but we did not write hint message for them. When there is not a hint message, the student can try as many times as they want to answer the problem and they are only told if they are wright or wrong; if a student is totally stuck they can hit a button and be told the answer. We hypothesize that just seeing the answer will not help the student learn and hint messages will most likely be helpful. However, some studies have shown that certain hint messages may not always be helpful.

Skill builders are the second components of the [44] study. Teachers could choose to assign from over 200 skill builders that ranged in skills from adding whole numbers to quadratic equation solving. A skill builder gives students practice on a topic until they get three problems right in a row. All skill builders where built at WPI and every problem had hint message.

One of the teachers who participated in the large-scale evaluation inspired TeacherASSIST. Mr. Chris LeSiege, a teacher from Gorham, Maine, considered on-demand assistance to be of utmost important to his students' success. For the duration of the study, he added hint messages to most problems from his textbook. Unfortunately we did not anticipate that teachers would do this, there was no way for other teachers using the same textbook to review what Mr. LeSiege created and adopt it for their classrooms. At the time of the study, on-demand assistance was considered a component of the problem, thus, only the owner of the problem could edit or add on-demand assistance. For every problem he did not own, in particular the textbook problems created at WPI, Mr. LeSiege had to manually create his own versions of the problem and write the on-demand assistance on his version. His valiant efforts left us with two versions of the questions and more interestingly a larger question of how we should move forward as an educational platform. First, how can we better facilitate enthusiastic and diligent teachers like Mr. LeSiege. Second, since Mr. LeSiege spent a tremendous amount of time and effort to create hint messages, could we use them to help not only support his students but also all the other students working on the same problems? And how effective would they be for students outside of his classrooms?

Several studies shown that on-demand assistance created by experts increased learning outcomes [33, 41, 4, 50, 5, 21]. However, some studies suggested that on-demand assistance may not always be beneficial. For instance, assistance that is too detailed or provides too much information could result in less learning gain [22, 24, 49]. In addition, consistency of tones and pedagogical strategies could plays an important role in learning. [26], giving advice for those authoring textbooks, suggested that it is important to ”establish a consistent standard.” Lack of consistency, especially in difficult topics, could cause learners to miss important connections between terms and concepts across multiple learning materials. Thus its not obvious how effective crowd-sourced assistance would be?

The idea of using crowd-sourcing in K-12 education is not new. For example, Teachers Pay Teachers (teacherspayteachers.com) allows teachers to buy and sell their lesson plans and teaching materials. In fact, the 2019 American Instructional Resources Surveys showed that 56% of American Math teachers used resources from Teachers Pay Teachers [40]. Several educational researchers such as [55] and [53] also created proof-of-concept systems that crowd-sourced learning materials from MTurk workers and re-distributed them to MTurk workers/learners. They found that the learning gain from the best crowd-sourced materials was comparable to the learning gain from materials created by experienced instructors. Crowd-sourcing has also been used to accomplish other tasks in learning systems. For example, DALITE [6] and Ripple Learning (ripplelearning.org) crowdsourced instructions and resources generated by their peers. PeerWise crowdsourced multiple-choice questions from learners and re-distributed them to their peers [11]. Crowdsourcing had also been used to bring grading to scale such as [27], which is especially important in MOOCs. In fact, EdX, on of the most popular MOOC providers, is also a good example of how to use crowdsourcing to bring online MOOCs to scale.

While there are many examples of using crowdsourcing in educational platforms, to our knowledge, there has yet to be an example of a live system that crowd-sources contents from real teachers and redistribute this directly to real students without teacher intervention, AND that reliably improves student learning.

In this work, we designed and implemented a feature called TeacherASSIST inside ASSISTments to gather on-demand assistance by using crowd-sourcing. This component would later re-distributed crowd-sourced on-demand assistance to students outside of creators' classes. TeacherASSIST was created to answer our three research questions:

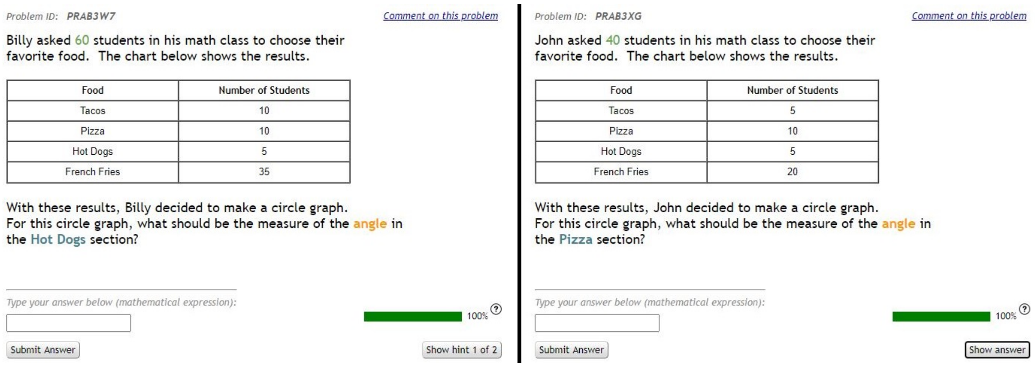
1. RQ1: "How could we design and implement a crowd-sourcing system that allows teachers to quickly and conveniently create on-demand assistance for their students?"

2. RQ2: "How effective is such crowd-sourced assistance?"

3. RQ3: "Could we reproduce the same result if the same randomized controlled trial is run in a different academic term?"

6.2 Background

In this work, we used ASSISTments an online tool used by teachers to support homework. ASSISTments (https://www.ASSISTments.org/) is a free online learning platform designed to empower teachers in their classrooms by automating laborious bookkeeping [19]. ASSISTments provides a library of problems, the majority of which is K-12 mathematics, that teachers can simple find, select, and assign to their students. ASSISTments provide immediate feedback as students work on their assignments and



Problem ID: PRAB3W7

Billy asked 60 students in his math class to choose their favorite food. The chart below shows the results.

Food	Number of Students
Tacos	10
Pizza	10
Hot Dogs	5
French Fries	35

With these results, Billy decided to make a circle graph.
For this circle graph, what should be the measure of the angle in the Hot Dogs section?

Problem ID: PRAB3XG

John asked 40 students in his math class to choose their favorite food. The chart below shows the results.

Food	Number of Students
Tacos	5
Pizza	10
Hot Dogs	5
French Fries	20

With these results, John decided to make a circle graph.
For this circle graph, what should be the measure of the angle in the Pizza section?

Figure 6-1: (left) For this problem on-demand assistance (hint messages) is available. Thus, the student may click ”Show hint 1 of 2” to request for hint messages. If no assistance is available (right), the student will only see ”Show answer” which would mark the student as having given up on the problem.

actionable reports to teachers. For every problem students receive instant correctness feedback, which tell the student whether the submitted answer is correct or not [17]. ASSISTments can also provide students with on-demand assistance, or ”tutoring.” Contrary to instant correctness feedback, on-demand assistance does not react to student answers. Rather, this type of assistance provides additional useful information and resources that help student solve the problem when requested (Figure 6-1). Both types of assistance have been shown to reliably improve student learning [41, 19, 43, 56, 29]. There are many types of on-demand assistance that had been shown to improve student learning such as step-by-step hints [19, 41], worked examples [15, 29], erroneous examples [29, 1], and providing the full solution to the problem [55, 53].

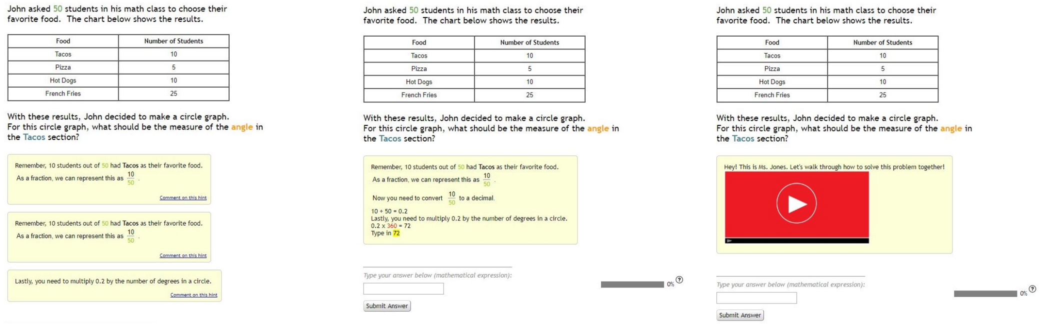
While many studies suggested that well-curated assistance improved student learning, there are also studies suggesting that some assistance may not be beneficial. A comprehensive literature review on the specificity of feedback and hint messages concluded that the literature is inconclusive on how specific feedback should be [49]. [24] showed that feedback with more information had a smaller effect on students’ ability to correct their own errors than feedback with less information, such as providing only the correct answer. Another meta-analysis suggested that more-detailed feedback could result in worse learning outcome [22]. In addition, since most instances of on-demand assistance in studies were created by either experts in learning fields

[33, 4, 50, 5] or by the instructors themselves [21], it would be dangerous to assume the same results for crowd-sourced on-demand assistance. In addition, since crowd-sourced assistance was created by neither experts nor their teachers, it is possible that the assistance could be of different tone or pedagogical strategies from those of the teachers or curricula. This inconsistency could reduce the effectiveness of learning materials and cause confusion [26].

There are several proof-of-concept studies on effectiveness of crowd-sourcing learning materials. For example, [53] crowd-sourced video lessons from MTurk workers and found that the learning gain from best crowd-sourced video was comparable to the learning gain from a popular video lesson from Khan Academy. Another system called AXIS [55] crowd-sourced explanations on how to solve a problem from MTurk workers. Then learners(other Mturkers were asked to revise and evaluate explanations as they solve problems. As learners work on problems, AXIS used machine learning to determine which explanations to present to to future learners. They found that explanations selected by AXIS were comparable to ones generated by experienced instructors, but all of this was done with Mturkers, not in authentic classrooms. To our knowledge, there is no live system that actively gets crowd-sourced assistance from teachers and directly redistribute them to students.

6.3 Methodology

Before we designed and implemented the crowd-sourcing system for RQ1, we first investigated how to incentivize teachers to create on-demand assistance and designed an algorithm to distribute it. Then, we investigated the impact of crowd-sourced on-demand assistance on student learning. In this work, all the implementations, data collection, and analysis were done inside ASSISTments , our methodology is not platform-specific and should be applicable to other online learning platforms of similar characteristics and features.



John asked 50 students in his math class to choose their favorite food. The chart below shows the results.

Food	Number of Students
Tacos	10
Pizza	5
Hot Dogs	10
French Fries	25

With these results, John decided to make a circle graph. For this circle graph, what should be the measure of the angle in the Tacos section?

Remember, 10 students out of 50 had Tacos as their favorite food.

As a fraction, we can represent this as $\frac{10}{50}$.

Remember, 10 students out of 50 had Tacos as their favorite food.

As a fraction, we can represent this as $\frac{10}{50}$.

Lastly, you need to multiply 0.2 by the number of degrees in a circle.

John asked 50 students in his math class to choose their favorite food. The chart below shows the results.

Food	Number of Students
Tacos	10
Pizza	5
Hot Dogs	10
French Fries	25

With these results, John decided to make a circle graph. For this circle graph, what should be the measure of the angle in the Tacos section?

Remember, 10 students out of 50 had Tacos as their favorite food.

As a fraction, we can represent this as $\frac{10}{50}$.

Now you need to convert $\frac{10}{50}$ to a decimal.

10 ÷ 50 = 0.2
Lastly, you need to multiply 0.2 by the number of degrees in a circle.
0.2 x 360 = 72
Type in 72

Type your answer below (mathematical expression):

John asked 50 students in his math class to choose their favorite food. The chart below shows the results.

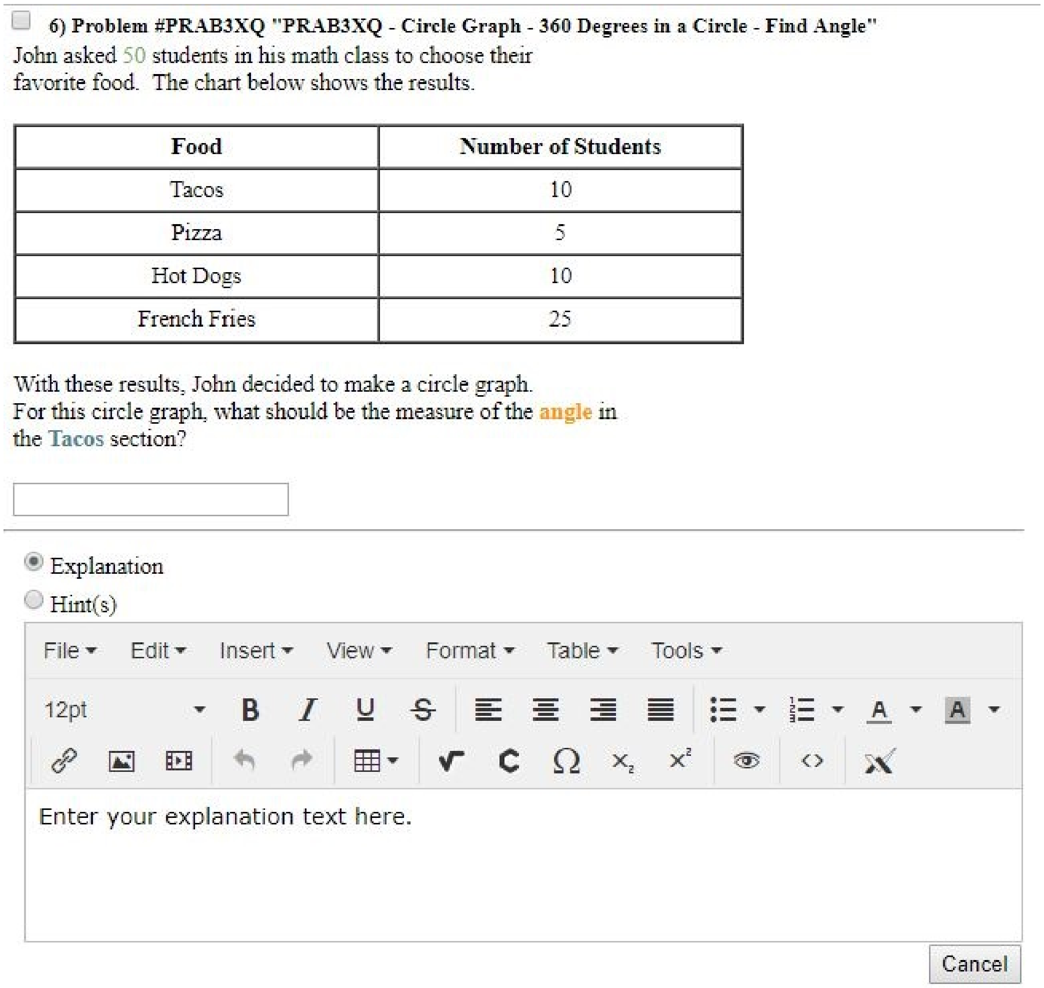
Food	Number of Students
Tacos	10
Pizza	5
Hot Dogs	10
French Fries	25

With these results, John decided to make a circle graph. For this circle graph, what should be the measure of the angle in the Tacos section?

Hey! This is Ms. Jones. Let's walk through how to solve this problem together!

Type your answer below (mathematical expression):

Figure 6-2: Examples of how the students see hints (left) and explanation (middle and right) in the ASSISTments tutor. Each yellow box in the left image represent a hint in the series. Explanations can be non-personal (middle) or personal (right).



☐ **6) Problem #PRAB3XQ "PRAB3XQ - Circle Graph - 360 Degrees in a Circle - Find Angle"**

John asked 50 students in his math class to choose their favorite food. The chart below shows the results.

Food	Number of Students
Tacos	10
Pizza	5
Hot Dogs	10
French Fries	25

With these results, John decided to make a circle graph. For this circle graph, what should be the measure of the angle in the Tacos section?

● Explanation
○ Hint(s)

Enter your explanation text here.

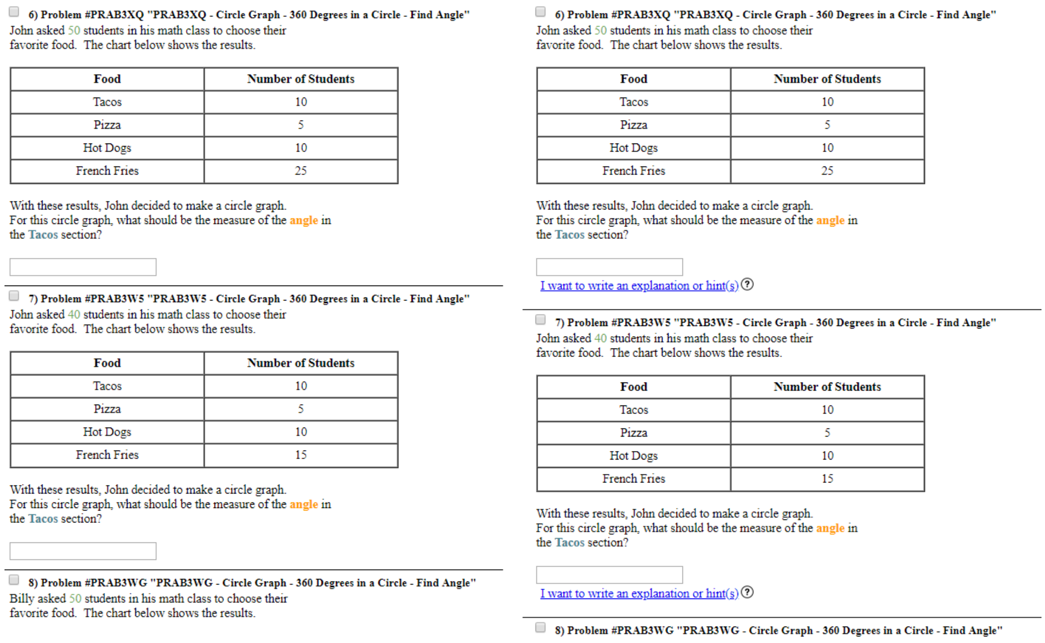
Figure 6-3: Teachers can choose to create a set of hints or an explanation for any problems of their choice.

6.3.1 Crowd-Sourcing On-Demand Assistance

For crowd-sourcing to be effective, we needed to obtain good quality on-demand assistance. The results of [55] shown that, given enough number of crowd-sourced on-demand assistance, we can obtain on-demand assistance of quality similar to one created by subject-matter experts. Thus, our goal was to design the system such that it is easy for teachers to create as much on-demand assistance as possible, as most users may not be motivated to contribute. However, as one of the main focus of ASSISTments and LMSs in general is to free teachers from laborious tasks, it is also important to not increase teachers' workload any more than needed. Thus, we collaborated with several teachers and investigated their normal everyday routines. The goal is to find the best approach to crowd-source on-demand assistance that are both convenient and beneficial to teachers' established routines for their classes and students.

The approach we took was first to create a component called "TeacherASSIST" inside ASSISTments. TeacherASSIST is a component allowed teachers to create on-demand assistance for their students as they taught the classes. Specifically, as teachers browsed through practice materials to assign to their students, they had an option to add their own on-demand assistance to each individual problem. This approach had many advantages. Firstly, teachers were incentivized to create on-demand assistance since it would directly benefit their students. Secondly, teachers were presented with the option to create on-demand assistance only for the problems they considered assigning to their students, so as not to overload them with too much to do. Lastly, the on-demand assistance was guaranteed to be of decent quality, as they belonged to the topics that teachers were currently teaching. Our implementation of TeacherASSIST was shown in Figure 6-4.

We then investigated what types of on-demand assistance should be supported. While we wanted to give teachers as much flexibility as possible, giving too many choices to the them could be detrimental and distracting [47]. We investigated the three types of on-demand assistance which were commonly available inside ASSIST-



□ 6) Problem #PRAB3XQ "PRAB3XQ - Circle Graph - 360 Degrees in a Circle - Find Angle"
John asked 50 students in his math class to choose their favorite food. The chart below shows the results.

Food	Number of Students
Tacos	10
Pizza	5
Hot Dogs	10
French Fries	25

With these results, John decided to make a circle graph.
For this circle graph, what should be the measure of the angle in the Tacos section?

□ 7) Problem #PRAB3W5 "PRAB3W5 - Circle Graph - 360 Degrees in a Circle - Find Angle"
John asked 40 students in his math class to choose their favorite food. The chart below shows the results.

Food	Number of Students
Tacos	10
Pizza	5
Hot Dogs	10
French Fries	15

With these results, John decided to make a circle graph.
For this circle graph, what should be the measure of the angle in the Tacos section?

□ 8) Problem #PRAB3WG "PRAB3WG - Circle Graph - 360 Degrees in a Circle - Find Angle"
Billy asked 50 students in his math class to choose their favorite food. The chart below shows the results.

□ 6) Problem #PRAB3XQ "PRAB3XQ - Circle Graph - 360 Degrees in a Circle - Find Angle"
John asked 50 students in his math class to choose their favorite food. The chart below shows the results.

Food	Number of Students
Tacos	10
Pizza	5
Hot Dogs	10
French Fries	25

With these results, John decided to make a circle graph.
For this circle graph, what should be the measure of the angle in the Tacos section?

I want to write an explanation or hint(s) ?

□ 7) Problem #PRAB3W5 "PRAB3W5 - Circle Graph - 360 Degrees in a Circle - Find Angle"
John asked 40 students in his math class to choose their favorite food. The chart below shows the results.

Food	Number of Students
Tacos	10
Pizza	5
Hot Dogs	10
French Fries	15

With these results, John decided to make a circle graph.
For this circle graph, what should be the measure of the angle in the Tacos section?

I want to write an explanation or hint(s) ?

□ 8) Problem #PRAB3WG "PRAB3WG - Circle Graph - 360 Degrees in a Circle - Find Angle"

Figure 6-4: The interface where teachers find and assign a subset of problems inside a problem set without (left) and with (right) the option to create on-demand assistance for their students

ments: hints, step-by-step problem-solving, and worked examples.

1. Hints are a series of helpful messages that provide students with some information they need in order to solve a problem. Hints are usually given to students one at a time when requested. This means after students see each hint, they can attempt to solve the problem right away to show that they've learned the materials. Many systems take away a portion of partial credits if they request for hints.

2. Step-by-step problem solving or ”scaffolding” problems is a type of on-demand assistance that breaks the original problems into smaller steps. The system will walk the students through each smaller step until the students reach the final ”step” problem, which answers the original problem. This allows students with low prior knowledge or struggling students to learn how to solve complicated problems by filling their missing knowledge as they work on scaffolding problems. [41].

3. Worked examples provide full explanations on how to solve the similar problems,

and sometimes the problem itself, from the beginning to the final answer. This type of on-demand assistance is analogous to teachers teaching students how to solve problems by demonstration.

We interviewed several teachers and educational researchers to find out the advantages and disadvantages of different types of on-demand assistance. In our final design, TeacherASSIST only allowed teachers to create hints and explanations, and not scaffolding problems. Creating scaffolding problems was complicated and time consuming, which is at odds with the narrative that teachers quickly create on-demand assistance as they assign problems to their students. In addition, even when the original problem is broken into smaller sub-problems, it is not uncommon for teachers to find struggling students stuck inside the ”step” problems due to knowledge gaps.

The other two types of on-demand assistance, hints and explanations, have different advantages and disadvantages. On one hand, many teachers expressed that explanations were the easiest and fastest to create, as they had already been doing it while teaching. On the other hand, many educational researchers and teachers preferred hints to explanations since hints allowed students to demonstrate learning within a problem. However, teachers reported that it was harder to create hints in many topics without giving away the answer itself. It is also important to note that on-demand assistance is not limited to text; teachers were also allowed to include images, tables, and any types of formatting (Figure 6-3) and multimedia such as videos (Figure 6-2).

6.3.2 On-Demand Assistance Distribution

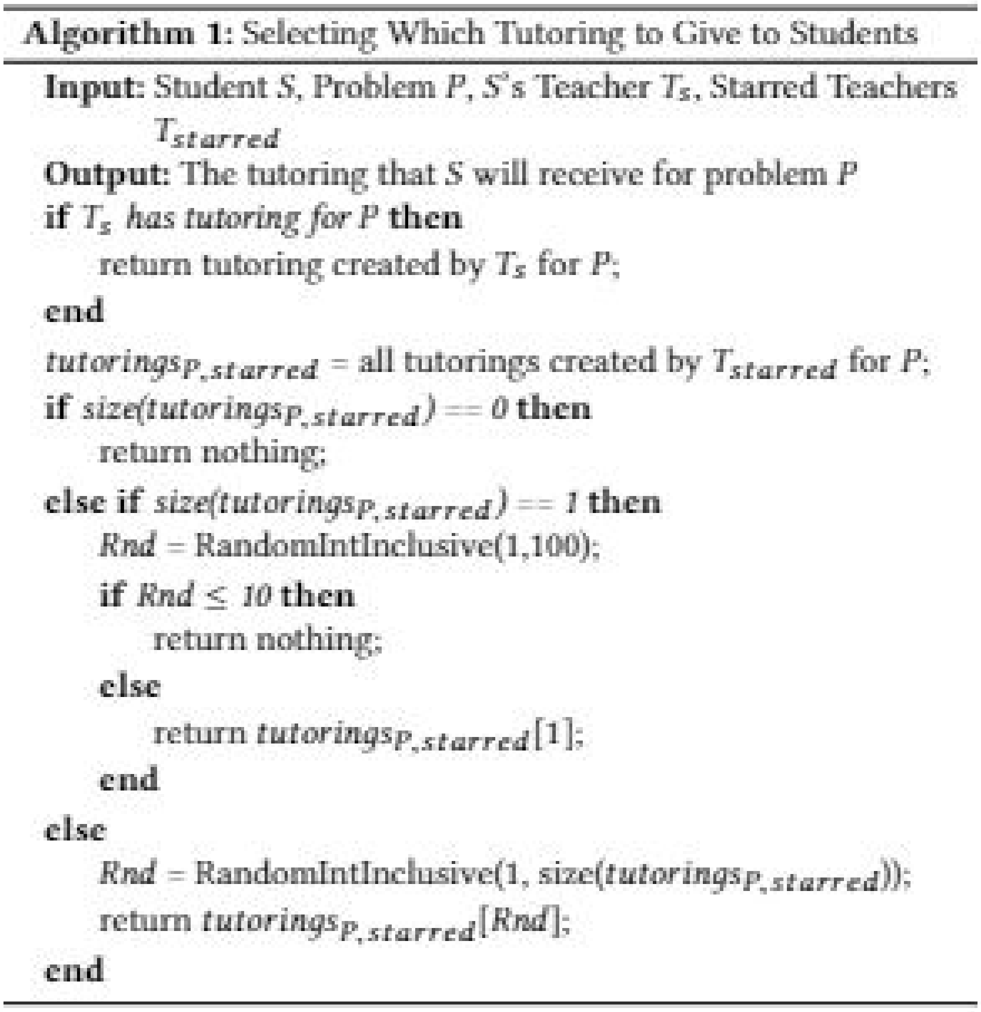
Before we distributed on-demand assistance, there were three major concerns we had to address. The first concern was privacy. While many teachers would not hesitate to create on-demand assistance for their own students, not as many felt comfortable sharing their on-demand assistance to students outside of their classes, especially if they included videos of themselves. Many teachers may not want to use on-demand assistance created by other teachers due to a different approach to solve the problems, which was the second concern. Lastly, as educational researchers, we wanted to be able to measure the quality of crowd-sourced on-demand assistance and to understand why each type of support suited different students through randomized controlled experiments.

In addition to the three concerns, there were three additional requirements that we considered to be most important. First, we needed to ensure that, if the teachers created on-demand assistance, their students must be guaranteed to receive them, regardless of what kinds of experiments were running and which other on-demand assistance is available. Second, since our main goal was to help students by providing them on-demand assistance as they are working on their assignments, it was important that such on-demand assistance be given out to as many students as possible. Third, we wanted to maintain the ability to conduct randomized control trials improve content as well as better on-demand assistance strategies.

As a result, we chose an approach similar to how new users in Wikipedia are promoted into confirmed and extended confirm users based on their activities [54]. For regular teachers, they can create any on-demand assistance for any problems. To address the first and second concern, such on-demand assistance will only be available to students in their own classes. Of those teachers, we searched for teachers who had regularly created on-demand assistance for their students and corrected any mistakes they found. With their consent, TeacherASSIST would re-distributed on-demand assistance created by starred teachers to students outside of their classrooms. This allowed us to scale-up on-demand assistance, addressing our second requirement. In order to satisfy the remaining concern and requirements, we came up with the distribution algorithm (Figure 6-5) that could run randomized controlled trials to determine the effectiveness of starred teachers' on-demand assistance.

6.3.3 Randomized Controlled Trials

TeacherASSIST was deployed in December 2017. We started promoting teachers to starred teachers in June 2018. Five teachers were promoted to starred teachers in 2018. Afterward, we started distributing starred teachers' on-demand assistance on



Algorithm 1: Selecting Which Tutoring to Give to Students

```
Input: Student $S$, Problem $P$, $S$'s Teacher $T_s$, Starred Teachers
        $T_{starred}$
Output: The tutoring that $S$ will receive for problem $P$
if $T_s$ has tutoring for $P$ then
    return tutoring created by $T_s$ for $P$;
end
$tutorings_{P,starred}$ = all tutorings created by $T_{starred}$ for $P$;
if $size(tutorings_{P,starred}) == 0$ then
    return nothing;
else if $size(tutorings_{P,starred}) == 1$ then
    $Rnd$ = RandomIntInclusive(1,100);
    if $Rnd \leq 10$ then
        return nothing;
    else
        return $tutorings_{P,starred}[1]$;
    end
else
    $Rnd$ = RandomIntInclusive(1, size($tutorings_{P,starred}$));
    return $tutorings_{P,starred}[Rnd]$;
end
```

Figure 6-5: The algorithm we used for selecting which on-demand assistance should be given to a student for a given problem.

October 10, 2018. The randomized controlled trial (named the "pilot experiment") started on the same date to answer RQ2. In 2019, we increased the number of starred teachers to nine and repeated the same randomized controlled trial (named "the repeated experiment") again to answer RQ3.

Specifically, the pilot experiment was conducted from August 9, 2018 to December 31, 2018 (corresponding to fall term of 2018). In this experiment, we compared crowd-sourced on-demand assistance (experimental condition) to simply giving the student the answer (control condition). For each problem with crowd-sourced on-demand assistance, the students were randomly assigned to one of the conditions at the problem-level. In other words, students could be in the control group for one problem, and in the experimental group for the next problem. We decided to use 9:1 as the ratio between the experimental condition and the control condition since we wanted to provide assistance to as many students as possible, and similar published works have shown similar on-demand assistance increases student learning. The repeated experiment was conducted and analyzed in the exact same manner as the pilot experiment, except it was conducted from January 1, 2019 to September 30, 2019 (corresponding to spring term and summer term of 2019).

When students worked on their assigned problems inside ASSISTments, they could see if there were on-demand assistance available before they requested it as seen in Figure 6-1. Specifically, if hint messages are available, students would see a button labeled "Show hint X of Y," where Y is the total number of hint messages available and X denotes which hint message will be given next. If no on-demand-assistance is available, the "Show answer" button will be displayed instead. Thus, we could not choose to analyze only students who requested for on-demand assistance since every student experienced the difference between condition, i.e. different buttons and corresponding partial credit costs, before receiving the treatment (i.e. requesting for on-demand assistance). Instead, we must first analyzed all students assigned to the control conditions and the experimental condition regardless of whether they actually requested for the assistance or not (we called this "intention-to-treat analysis"). After we determine that the button difference does not cause students in two conditions to behave significantly differently, we would then be able to analyze only students who request for assistance in the experimental condition or the answer in the control condition (we called this "treated analysis").

In the following section, we refer to the problems of where crowd-sourced on-demand assistance appeared as "RCT problems," and the math problems that the students worked on immediately after the RCT problems as "next problems." It is important to note that, for different students, the next problems were not guaranteed to be the same. In fact, for some RCT problems, the next problems may be in a different assignment, worked on a different day by the student. We will also use the term "ask for help" to refer to both students requesting on-demand assistance (experimental condition) and students requesting for the answer (control condition).

In this work, we only analyzed data where both RCT problem and the next problem come from to the same assignment.

In order to measure the quality of crowd-sourced on-demand assistance, we looked at 4 next-problem dependent measures.

1. "next problem correct first try": did the students answer the next problem correctly on their first try without using assistance or asking for the answer?
2. "next problem ask for help": did the students request for assistance or the answer during the next problem?
3. "next problem stop out": did the students give up solving the next problem?
4. "next problem attempt count": the number of attempts the student made during the next problem.

Our hypothesis was that the crowd-sourced on-demand assistance improved students learning. Students should be able to correctly answer the next problems more and ask for help less as they no longer need them. We did not expect a single problem-solving session to drastically change stop out rate or next problem attempt count. These two measures were included in the analysis to ensure that the differences between the correctness and help usages in the control condition and the experimental

	number of problems solved	number of problems correctly solved on first try (percent)	number of problems where students requested for assistance or answer (percent)
teacher's own class	29049	19709 (67.84%)	4857 (16.72%)
control	13857	9377 (67.67%)	2271 (16.38%)
experimental	128153	86877 (67.79%)	20925 (16.32%)

Table 6.1: A table showing the availability and usages of teacher-created on-demand assistance and the crowd-sourced on-demand assistance.

condition, if detected, were not caused by one of the conditions causing students to disproportionately give up on the next problems.

6.4 Results

6.4.1 Overall Usage of TeacherASSIST

We investigated whether TeacherASSIST was able to incentivize teachers to create on-demand assistance. By the end of 2019-2020 academic year, three years after TeacherASSIST was deployed, we found that 146 different teachers had used TeacherASSIST to create 40,292 instances of on-demand assistance for 25,957 distinct problems across different curricula, 16,493 of which belong to our 9 starred teachers. Out of 146 teachers, 29 teachers had created more than 50 instances of assistance and 14 of those teachers created more than 1,000 instances of assistance over three years.

To put the number in perspective, in 2017-2018 academic year, 132,738 distinct problems were assigned inside ASSISTments, only 38,194 of which had non-TeacherASSIST on-demand assistance. Of those problems, 27,094 more instances of on-demand assistance were created through TeacherASSIST, increasing the number of on-demand assistance by 70%.

6.4.2 Pilot Experiment

To measure how effective crowd-sourced on-demand assistance was (RQ2), we analyzed logged data of students who received on-demand assistance. We obtained problem log data from ASSISTments. For the duration of pilot experiment, there were 1,795 instances of on-demand assistance created for 1,787 unique problems. Out of instances of 1,795 on-demand assistance, 1,546 were explanations and 248 were hints. There were 142,010 problems solved in the randomized controlled trial, 128,153 of which received crowd-sourced teacher on-demand assistance and 13,857 of which only the answer was available. Our dataset is publicly available here [37].

Availability and Usages

Table 6.1 shows the availability and usages of teacher-created on-demand assistance and the crowd-sourced on-demand assistance. We found no significant difference between the percentage of students in the control condition and the experimental condition who answered the RCT problems correctly on their first try without asking for on-demand assistance ($p < 0.05$). Similarly, we found no significant difference between the percentage of students who requested crowd-sourced on-demand assistance (experimental) and students who requested for the answer (control) ($p < 0.05$).

Effects on Next Problems

To analyze the effects of crowd-sourced on-demand assistance on the next problems, we conducted the intention-to-treat (ITT) analysis. An intention-to-treat analysis is an analysis in which everyone who participated in the RCT is included in the analysis regardless of their scores, characteristics, and interaction with the intervention inside the RCT. Since our dataset was a problem-student level (i.e., a log of a student solving a problem), each observation was not independent (because one student solved multiple different problems and one problem was solved by multiple different students). Using t-test directly on the problem-student level would violate the independence observation assumption of t-test. Instead, we aggregated observa-

next problem dependent var.	ctrl mean	exp. mean	t- stat	p- value	corrected p-value
correct first try	0.65	0.66	-0.86	0.39	0.74
ask for help	0.17	0.16	0.86	0.39	0.74
stop out	0.03	0.03	0.48	0.63	0.74
attempt count	1.53	1.52	0.33	0.74	0.74

Table 6.2: Pilot Experiment: problem-level paired t-test intention-to-treat analysis on student next problem dependent variables using t-test and Benjamini–Hochberg. The number of unique problems = 1293

tions into 1) problem-level and 2) student-level, applied paired t-test on the aggregated observations, and reported the result of both aggregation methods for both the intention-to-treat and treated analysis. Since we performed multiple t-tests, we used the Benjamini–Hochberg procedure to obtain corrected p-values to reduce false positive.

In addition to the intention-to-treat analysis, we also looked at treated analysis. Treated analysis, in contrast with ITT analysis, only looks at participants who interact with the intervention or treatment. In our work, the treated analysis means that we would only look at students who asked for help while they worked on the RCT problem. The reason we also conducted the treated analysis was because a large majority of the students (67%) in both conditions were able to answer the RCT problems on their first try without requesting any on-demand assistance. In addition, only a small portion of the students (16.7%) asked for help. This means the main difference between conditions (crowd-sourced on-demand assistance vs. answer) could be observed only on a small fraction of the students. Thus, in order to detect the effects in ITT analysis, the effects of the on-demand assistance must be very large to avoid being overshadowed by most of the samples that were not treated.

Intention-to-Treat Analysis

Table 6.2 and 6.3 shows the problem-level and student-level intention-to-treat analysis of the effect of crowd-sourced on-demand assistance using paired t-test. We found no significant difference between any next problem dependent measures using 5%

next problem dependent var.	ctrl mean	exp. mean	t- stat	p- value	corrected p-value
correct first try	0.63	0.63	-1.35	0.18	0.23
ask for help	0.18	0.17	2.31	0.02	0.08
stop out	0.03	0.03	-0.81	0.42	0.42
attempt count	1.57	1.53	1.90	0.06	0.11

Table 6.3: Pilot Experiment: student-level paired t-test intention-to-treat analysis on student next problem dependent variables using t-test and Benjamini–Hochberg. The number of unique students = 4181

false positive rate (alpha = 0.5) for Benjamini–Hochberg procedure. This is expected according to since Table 6.1 shows that, of all logged solved problems, more than 60% of the time students were able to solve the problems correctly on their first attempt without using on-demand assistance. In addition, students requested for on-demand assistance less than 20% of the time. In another word, a large majority of the students did not experience the difference between the control condition and the experimental condition.

Treated Analysis

Table 6.4 and Table 6.5 show the paired t-test of problem-level and student-level treated analysis of the effect crowd-sourced on-demand assistance. We found that, after applying Benjamini–Hochberg procedure, students who saw the on-demand assistance were less likely to request for more on-demand assistance in the next problem with statistical significance (corrected p-value < 0.01). This result can be interpret as either a positive or a negative effect of crowd-sourced on-demand assistance on learning. Students may either 1) learned enough to be able to solve the next problem, thus additional on-demand assistance was not needed, or 2) did not feel like on-demand assistance helps (e.g. of poor quality) and decided that requesting for any more on-demand assistance was not worth the partial credit cost. Using only the result data from the pilot experiment, we hypothesize that it was more like that crowd-sourced on-demand assistance had a positive impact on learning since, in addition to being well-supported by literature, while not statically significant, the percent of students

next problem dependent var.	ctrl mean	exp. mean	t- stat	p- value	corrected p-value
correct first try	0.39	0.40	-0.80	0.42	0.42
ask for help	0.46	0.43	2.25	0.02	0.10
stop out	0.03	0.03	-0.85	0.40	0.42
attempt count	1.86	1.91	-1.04	0.30	0.42

Table 6.4: Pilot Experiment: problem-level paired t-test treated analysis on student next problem dependent variables using t-test and Benjamini–Hochberg. The number of unique problems = 620

next problem dependent var.	ctrl mean	exp. mean	t- stat	p- value	corrected p-value
correct first try	0.39	0.41	-1.50	0.13	0.23
ask for help	0.45	0.41	3.39	<0.01	<0.01
stop out	0.03	0.04	-0.47	0.64	0.64
attempt count	1.91	1.85	1.35	0.18	0.23

Table 6.5: Pilot Experiment: student-level paired t-test treated analysis on student next problem dependent variables using t-test and Benjamini–Hochberg. The number of unique students = 1256

in the experimental condition who answered their problem correctly on their first try is higher than that of the control, as well as with slightly lower attempt count.

6.4.3 Repeated Experiment

Using our data from the pilot study, we hypothesize that crowd-sourced on-demand assistance was of acceptable quality to improve student learning, causing them to answer more problems correctly while requiring less additional on-demand assistance. From January 1, 2019 to September 30, 2019, there were 232,248 problems solved in the randomized controlled trial, 208,987 of which received crowd-sourced teacher on-demand assistance and 23,261 of which only the answer was available. In said solved problems, 3,515 unique problems were solved with 3,698 distinct instances of on-demand assistance. Out of said on-demand assistance, 2,475 were explanations and 1,222 were hints. Similar to the pilot study, we found no significant difference between the percentage of students in the control condition and the experimental condition who answered the RCT problems correctly on their first try without asking

next problem dependent var.	ctrl mean	exp. mean	t- stat	p- value	corrected p-value
correct first try	0.62	0.63	-2.29	0.02	0.04
ask for help	0.20	0.19	3.65	<0.01	<0.01
stop out	0.03	0.02	1.47	0.14	0.19
attempt count	1.60	1.58	0.85	0.39	0.39

Table 6.6: Repeated Experiment: problem-level paired t-test intention-to-treat analysis on student next problem dependent variables using t-test and Benjamini–Hochberg. The number of unique problems = 2379

next problem dependent var.	ctrl mean	exp. mean	t- stat	p- value	corrected p-value
correct first try	0.65	0.65	-1.59	0.11	0.15
ask for help	0.17	0.16	1.65	0.10	0.15
stop out	0.02	0.02	1.14	0.25	0.25
attempt count	1.60	1.55	2.86	<0.01	0.02

Table 6.7: Repeated Experiment: student-level paired t-test intention-to-treat analysis on student next problem dependent variables using t-test and Benjamini–Hochberg. The number of unique students = 6945

for on-demand assistance ($p > 0.05$). We also found no significant difference between the percentage of students who requested for crowd-sourced on-demand assistance (experimental) and students who requested for the answer (control) ($p > 0.05$).

Intention-to-Treat and Treated Analysis of the Repeated Experiment

Interestingly, several tests from the intention-to-treat analysis are statistically significant even after using Benjamini–Hochberg with alpha = 0.05. Specifically, Table 6.6 shows that when aggregated on the problem level, students in the experimental condition were more likely to answer the next problems correctly on their first attempt as well as asking for less on-demand assistance on their next problem than students in the control condition (corrected p-value = 0.04 and <0.01, respectively). In addition, Table 6.6 showed that when aggregated on the student level, students in the experimental condition were more likely to have a smaller number of attempts than students in the control (corrected p-value = 0.02). Since inside ASSISTments, students were required to answer the problem correctly before they could move on to the next problem, a lower number of attempts meant students reached the correct answer faster on average, given there was no change in other dependent measures.

As for the treated analysis, the result aligned with the result of our pilot experiment. Table 6.8 and Table 6.9 show that the students in the experimental conditions asked for less on-demand assistance in the next problem (corrected p-value = 0.04 and correct p-value < 0.01 for problem-level and student-level aggregation, respectively). While not statistically significant, students in the experimental condition were more likely to answer the next problems correctly on their first attempt as well as asking for less on-demand assistance on their next problem than students in the control condition similar to the results we obtained from the pilot study and the intention-to-treat analysis.

6.5 Conclusion

In this work, we designed and implemented a mechanism that allows online learning platforms to crowd-source on-demand assistance from teachers. We developed this scheme in close collaboration with teachers and educational researchers to ensure that it is both convenient and beneficial to teachers, while remain open enough for researchers to conduct meaningful research.

To answer RQ1, we interviewed teachers and subject-matter experts to find out what are the features and requirements expected of on-demand assistance crowd-sourcing system, TeacherASSIST. Teachers wanted the system to improve student learning without overtaxing them and without additional work. Educational researchers wanted to be able to investigate the effectiveness of different kinds of on-demand assistance. Our ability to conduct RCTs for RQ2 and RQ3 shown that researchers can use TeacherASSIST to investigate the effectiveness of different kinds of on-demand assistance. While TeacherASSIST was designed and implemented inside ASSISTments, the core design and algorithm are applicable to other platforms that support on-demand assistance and content creation. Originally, only 38,194 of 132,738 distinct problems assigned inside ASSISTments in 2017-2018 academic year

next problem dependent var.	ctrl mean	exp. mean	t- stat	p- value	corrected p-value
correct first try	0.36	0.37	-1.41	0.16	0.32
ask for help	0.49	0.47	2.54	0.01	0.04
stop out	0.03	0.03	0.01	1.00	1.00
attempt count	1.95	1.92	1.00	0.32	0.42

Table 6.8: Repeated Experiment: problem-level paired t-test treated analysis on student next problem dependent variables using t-test and Benjamini–Hochberg. The number of unique problems = 1312

next problem dependent var.	ctrl mean	exp. mean	t- stat	p- value	corrected p-value
correct first try	0.38	0.40	-1.86	0.06	0.08
ask for help	0.46	0.42	3.25	<0.01	<0.01
stop out	0.03	0.03	-0.68	0.50	0.50
attempt count	2.06	1.97	2.15	0.03	0.06

Table 6.9: Repeated Experiment: student-level paired t-test treated analysis on student next problem dependent variables using t-test and Benjamini–Hochberg. The number of unique students = 1955

had on-demand assistance. By the end of 2019-2020 academic year, 27,094 instances of on-demand assistance were created for those problems through TeacherASSIST, starred teachers and otherwise. When we looked outside of the 2017-2018 dataset, we found a total of 40,292 instances of on-demand assistance across 25,957 distinct problems in different curricula, 16,493 of which belong to our 9 starred teachers. We also found that 14 teachers used TeacherASSIST heavily, creating more than 1,000 instances of assistance over three years.

To answer RQ2, we conducted the pilot RCT from August 9, 2018 to December 31, 2018. We found that students who requested the crowd-sourced on-demand assistance were reliably less likely to require additional assistance in the next problem. While the effect was small, it was expected since the experiment was conducted on the problem-level. Students who requested on-demand assistance were also more likely to correctly answer the next problem on their first attempt with lower overall average number of attempts, though it was not statistically significant.

To answer RQ3, we repeated the experiment we ran in RQ2 during the following academic term from January 1, 2019 to September 30, 2019. The results of the repeated experiment was in the same direction as RQ2, further confirm our hypothesis that crowd-sourced on-demand assistance is of high quality enough to improve student learning.

We concluded that we think the future of crowd-sourcing is bright. While there are several other crowd-sourcing applications in education such as [55] and [53], we are the first to crowd-source directly from active users (K-12 teachers) and redistributed crowd-sourced contents in a live environment. Our work serves as an evident that teachers are willing and able to create and improve contents of learning management systems, given that such contents are helpful to their students. We believed that a major part of this success was due to the fact that the designed of TeacherASSIST was heavily focused on teachers' need; TeacherASSIST was nicely integrated into teachers' routine and the on-demand assistance will directly benefit both their current and future students.

We also published the anonymized dataset from our large-scale randomized control trials. In this dataset, we included the data from both our pilot and the repeated experiments. All logged data (intention-to-treat) were included.

The code we used for analysis and datasets can be found here. `https://doi.org/10.17605/OSF.IO/EGP5F`

6.6 Future Work

In this work, our analysis is limited to next problem analysis. Ideally, we would like to measure student learning e.g. by using pre-test and post-test. However, since our randomized controlled trial was on an individual problem-level, it was impossible for us to have a proper pre-test and post-test. In order to solve that, we plan to design and run a different randomized controlled trial that would allow us to have some control over what the next and previous problems are using the problem set structure.

Alternatively, we could measure student learning by using more history and "future" information. For instance, we could compare the students history 10 problem before and after the RCT to get a better estimate of student learning. We would like to also look at the effects of on-demand assistance over multiples consecutive RCT problems as opposed to a single RCT problem. We expect this approach to have significantly bigger effect on student learning that what we have shown in this work.

In term of scalability, our method to aggregate on-demand assistance is currently naive. With better aggregation methods, we believe that the system would be able to select a better on-demand assistance, causing better improvement in student learning.

Chapter 7

Improvement on Hint and Explanation Crowdsourcing Method for an Online Learning Platform

One of the most important advantages for using crowdsourcing was scalability. A well-implemented crowdsourcing feature allowed the platform to continuously scale with users and time with minimal intervention from the administrators. TeacherASSIST, as it stood, was able to grow the pool of student supports only by adding more starred teachers, which required administrator intervention. While, technically, the pool also grew as each individual starred teacher created student supports for their classes, there's a limit to how much it could grow, which was the curricula and teaching materials they're using.

By using the 4 stages mentioned in [18], I was able to identify where TeacherASSIST was lacking. TeacherASSIST, as it stood, lacked greatly in stage 2, 3, and 4. For stage 1, only starred teachers could be considered contributors as other teachers' student supports were only available to their students (i.e. not crowdsourced). There's no accessibility of peer contribution at all. The aggregation method was simple and naive. For the fourth stage, teachers got nothing back from creating more student supports and they weren't even informed of how much their student supports were used or seen by students.

In order to take full advantage of crowdsourcing, I proposed and implemented improvements on each of the 4 stages. For the first stage, a new feature called "trusted teachers" was implemented. This feature allowed teachers to designate other teachers as trusted. This, in turn, allowed their students to receive student supports created by anyone they designated as trusted teachers. This allowed teachers who use materials outside starred teachers' curricula to share and compliment each others' student supports.

For the second stage, I designed a prototype which would allow teachers to see each others' student supports. Teachers were asked to rate student supports they see. Theoretically, this kind of feature, when implemented, allows teachers to tailor student supports from various teachers to fit their classes and students. Practically, this process would consume so much time that only a tiny fraction of teachers would do it. So, instead, I treated this second stage as a data collection process to find out who were good teachers in comparison to existing starred teachers.

In addition, as TeacherASSIST learned of more well-rated teachers, it could start giving student supports based on such ranking. As a result, TeacherASSIST became better at aggregating student supports from various teachers, which was stage 3. I ran a randomized controlled trial to investigate the effect of such ranking with additional non-starred teachers.

Lastly, I designed prototype reports that informed teachers of how their student supports were used and how useful they were. These prototypes were meant to inform teachers of what kind of information could be available to them. The main goal was to notify them that what they created matter, in hope that they would continue to create more student supports, which would improve stage 4.

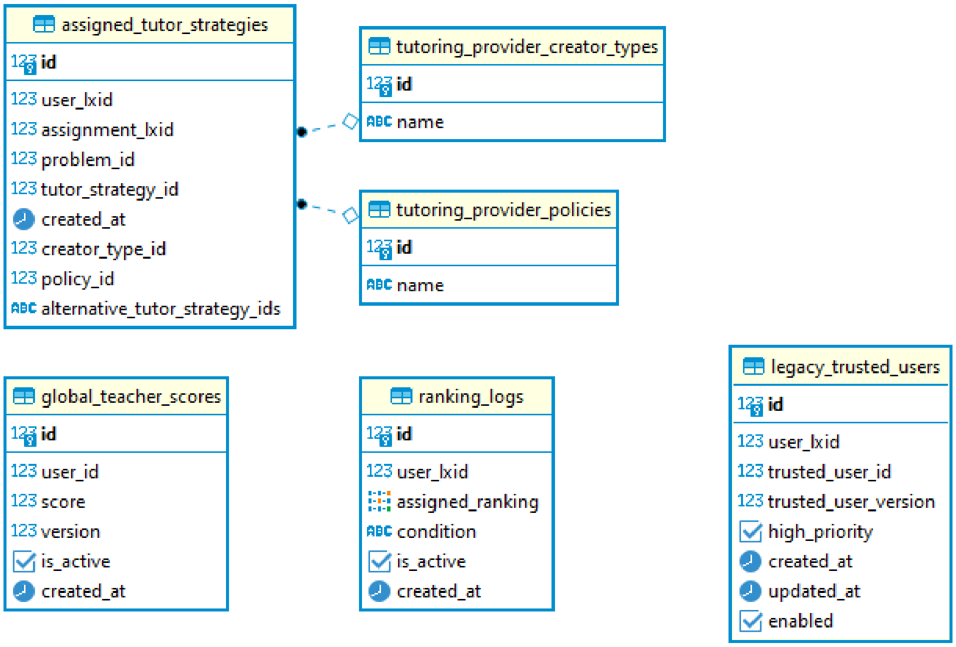


Figure 7-1: The ER Diagram of the tables that store TeacherASSIST-related information.

7.1 TeacherASSIST Infrastructure

In order for TeacherASSIST to support trusted teachers, its infrastructure and database schema needed to be updated. The updated database ER diagram is shown in figure 7-1.

Table tutoring_provider_creator_types and tutoring_provider_policies were previously a single table teachassist_policy. This set of tables was designed to store information on how student supports that were given to the students were selected and stored in assigned_tutor_strategies, such as randomly selected from a pool of starred teacher student supports. With a single table, combinations of policies and creator types have to be pre-specified in the database to be considered valid. This new approach decoupled policies and creator types, allowing new policy or creator type to be added without having to worry about the other. When TeacherASSIST distributed student supports to students, student supports could be separated into different priorities based on the creators/owners of student supports, namely the student's teachers, starred teachers, and trusted teachers which were added for this dissertation work. I defined this information ”creator type.”

Policies, on the other hand, stored which method was used to select a student support from all the student supports of the same priority that were available. For instance, when only one student support was available, I ran an RCT comparing that student support against the control in my previous work. After two successful replication studies in [36] and in 2020, there's no more need to run such RCT, allowing all students to receive student supports. When multiple student supports were available, the default policy was to randomly select one. For the RCT in this dissertation work, a student support was selected using the ranking according to student support scores.

In addition, this new separation of creator type and policy would also allow TeacherASSIST to easily integrate with 2 other features inside ASSISTments that were under development. One is E-TRIALS, a feature that would allow researchers to run randomized controlled trials inside ASSISTments. E-TRIALS needed policies such as ”preassigned during assigned time” and creator types such as ”researchers.” Another feature is the Reinforcement Learning Service (RLS). When multiple student supports were available, RLS would actively evaluate them using past student performance data as well as the current student data and identify which student support would be best for the student. RLS would need a new policy type ”RLS.”

Table legacy_trusted_users stored data regarding teachers who designated other teachers as trusted teachers, wanting to allow their students to receive said teachers' student supports. In addition, as ASSISTments was, at the time, in the process of migration from 1.0 infrastructure to 2.0 infrastructure as of 2021, this database also allowed students of teachers who moved to 2.0 to be able still receive student supports their teachers created in 1.0 before the official ASSISTments data migration.

Table global_teacher_scores stored data regarding the average scores of student supports of teachers. Table ranking_logs stored the mapping from the students to a ranking assigned to them (global in experimental condition, a random ranking for control). These two tables as well as the server code were designed to support versioning. This allowed the sets of scores and rankings to be updated as newer information, scores, or experiments were needed, while maintaining logs of previously used scores and ranking.

7.2 Trusted Teachers and Evaluation

For this work, I designed "trusted teachers" into TeacherASSIST. This new feature allowed teachers to designate other teachers as "trusted," allowing their students to receive student supports from such teachers. This feature would allow teachers to be able to organically grow the pool of student supports their students may receive without intervention or maintenance. Compared to trusted teachers, starred teachers may be considered globally trusted teachers.

In order for teachers to grow their list of trusted teachers, one natural way was to allow the teacher to see other teachers' student supports and mark the ones they'd want their students to receive. When the teacher designated enough of one teacher's student supports as preferable then the teacher would be "trusted." This process of becoming trusted may require explicit user inputs or happen automatically depending on what the system deemed appropriate, or by teacher's explicit requests. Incidentally, this process of curating student supports implicitly created a ranking of student support creators where supports by one creator was generally perceived as desirable by teachers more than supports created by others.

While this process was organic and theoretically scalable, it presented one more issue into TeacherASSIST: it created another crowdsourcing task that only benefitted teachers who had completed the task. And, like all crowdsourcing tasks, having low percent of contributors was inevitable. In other words, as it stands, this feature would only benefit a small subset of users if implemented.

Thus, I took the ranking one step further from a user-level to an aggregated, system-level ranking. In other words, this aggregated ranking presented how preferable student supports from each creator is compared to other creators. This global ranking could be used as a default "ranking" for teachers who may not have time to rate student supports to find their own trusted teachers. As a result, all teachers inside the system would be able to expand their pools of student supports for their students, whether through their own ranking or aggregated, system-wide ranking of teachers.

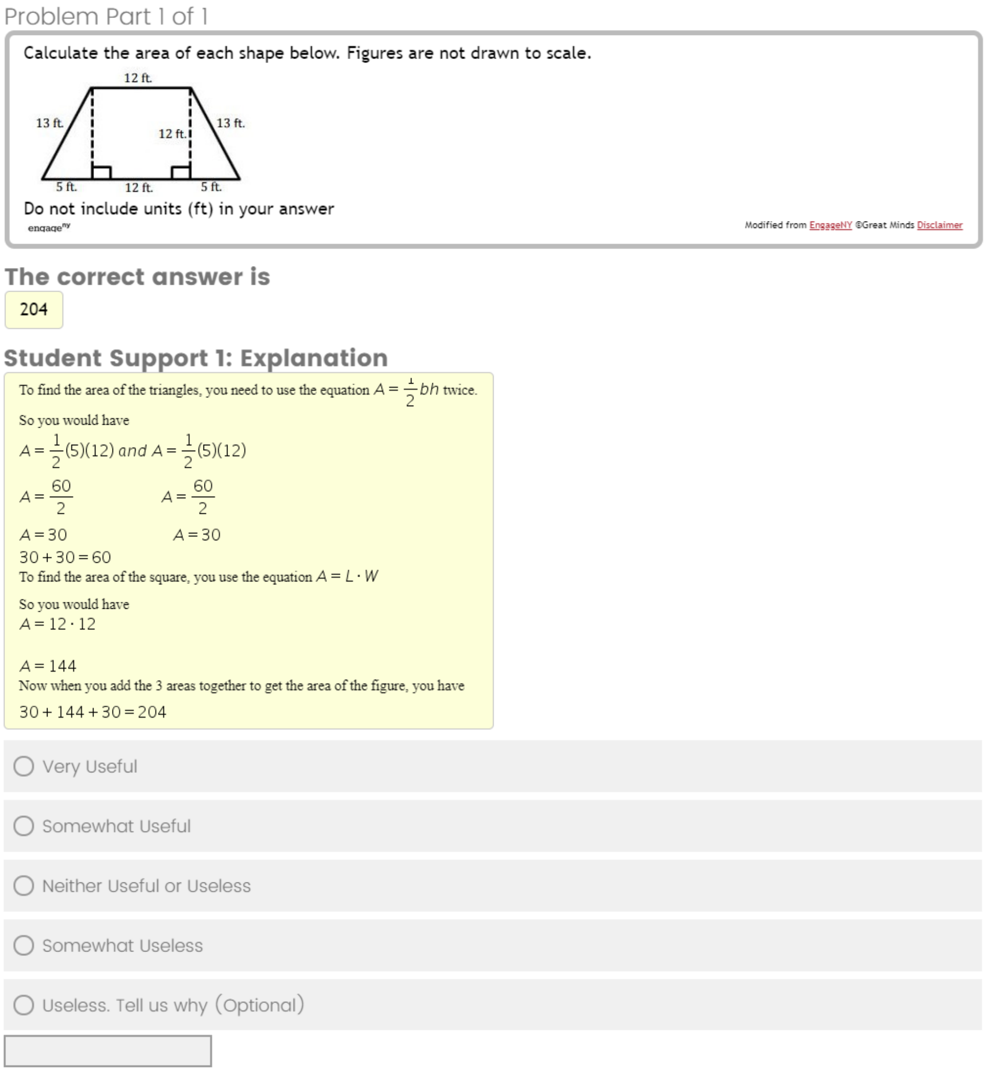
For this work, I created a proof-of-concept of the student support rating process using the Qualtrics survey tool and implemented a fully-functional delivery policy using global ranking. Then, I ran a randomized controlled trial in order to measure the effectiveness of the delivery policy using global ranking.

7.2.1 Student Support Rating System

The main goal of this proof-of-concept student support rating system was to be able to gather (crowdsource) teacher opinions on student supports created by various teachers. Since my end goal was to calculate an average score of each teacher across curricula and grades, this rating system must cover problems from all curricula and grades of interest, while ensuring that there were enough samples of each teacher to calculate reliable teacher scores.

For this work, I only considered Engage NY and Illustrative Mathematics, and 6th, 7th, and 8th grade as curricula and grades of interest, accordingly. This was because they were most commonly used inside ASSISTments and had the largest number of student supports created. Then, I selected all teachers who created more than 100 student supports across aforementioned curricula and grades. For each curriculum-grade, I randomly selected 16 problems that I would use in the student support rating survey. I used the following criteria in order to select problems from each curriculum-grade:

- Each selected problem must have at least 3 student supports created by 3 different teachers. - For problems with more than 3 student supports available, 3 were chosen such that the numbers of selected student supports of teachers within the curriculum-grade were as balanced as possible. - Each part ("sub-problem") of a multi-part problem counts as an individual problem. A sub-problem can be selected only if 1) the first part was also selected 2) the sub-problem only depends on the first part. For example, if a problem has 4 parts, only part 1, 3, and 4 (i.e. without part 2) may be selected as long as each satisfies all criteria. - The selected 16 problems must be separable into two sets of 8 problems with matching multi-part "signature" i.e. if one set contained a 4-part problem, 3-part problem, and a single problem (4



Problem Part 1 of 1

Calculate the area of each shape below. Figures are not drawn to scale.

Do not include units (ft) in your answer

The correct answer is

204

Student Support 1: Explanation

To find the area of the triangles, you need to use the equation $A=\frac{1}{2}bh$ twice.

So you would have

$A=\frac{1}{2}(5)(12)$ and $A=\frac{1}{2}(5)(12)$

$A=\frac{60}{2}$ $\quad$ $A=\frac{60}{2}$

$A=30$ $\quad$ $A=30$

$30+30=60$

To find the area of the square, you use the equation $A=L\cdot W$

So you would have

$A=12\cdot 12$

$A=144$

Now when you add the 3 areas together to get the area of the figure, you have

$30+144+30=204$

○ Very Useful

○ Somewhat Useful

○ Neither Useful or Useless

○ Somewhat Useless

○ Useless. Tell us why (Optional)

Figure 7-2: An example screenshot of the student support rating survey using the Qualtrics survey tool.

+ 3 + 1 = 8 problems), the other 8 problems must also be a 4-part problem, 3-part problem, and a single problem. - During Summer of 2019, ASSISTments hired several WPI students to create student supports for several curriculum-grades, including the curriculum-grade used in this study. For the purpose of this survey and ranking, I considered all such students a single user/creator as they all received the same instructions on how to create student supports under the same supervisor as well as how to collaborate enough to ensure that they have as little overlap as possible.

In order to create such a ranking, I created a survey using Qualtrics where volunteer teachers rate individual student support. In the survey, each teacher was presented with 8 semi-randomly chosen problems from a curriculum and grade of their choosing. Specifically, the 8 problems would have the signature previously used in the criteria of choosing problems e.g. if the curriculum-grade had 2 sets of 4-3-1 problems, one teacher may see the first 4-part problem, the second 3-part problem, and the first single problem. Another teacher may see the second 4-part, first 3-part, and second single part.

For each problem, the teacher was presented with the problem text, the correct answer, and a list of three student supports created by 3 different teachers. For each student support, the teacher was asked to rate it on a Likert scale 1-5: useless (1), somewhat useless (2), neither useful or useless (3), somewhat useful (4), and very useful (5). An example of a student support rating item in the survey is shown in figure 7-2.

The survey was sent out with Teachers For Research and Feedback (TFRF) ASSISTments newsletter on March 23, 2021, with a $150 lottery drawing as an incentive for those who completed the survey by March 31, 2021. There were a total of 27 responses, 2 of which were deemed invalid (all student supports were rated with the same rating). Then, I calculated the average rating for each teacher who created student supports shown in the survey. The average scores of teachers were shown in table 7.1.

	n	mean	standard error
starred teacher 1	49	4.1633	0.1734
starred teacher 2	46	4.2174	0.1671
starred teacher 3	27	4.2222	0.2285
starred teacher 4	53	3.6981	0.2182
starred teacher 5	102	4.1471	0.1265
non-starred teacher 1	9	4.0000	0.2887
non-starred teacher 2	20	4.5000	0.1147
non-starred teacher 3	45	4.2889	0.1334
non-starred teacher 4	71	4.0423	0.1396
non-starred teacher 5	23	4.1304	0.2615
hired students (combined)	155	4.0903	0.1061

Table 7.1: The average scores of each teacher included in the survey. n was the number of times their student supports were rated across all their student supports included in the survey.

7.2.2 Randomized Controlled Trial

In Spring, 2021, I ran a randomized controlled trial (RCT) in order to evaluate the effectiveness of the delivery policy using the global ranking. Students in the experimental condition received student supports based on the global ranking of teachers using data from the student support rating survey. In the control condition, students received student supports based on a randomized ranking of starred teachers. In this RCT, students were randomly assigned into conditions prior to the experiment and continued to receive student supports selected using their respective policies for the entire duration of the experiment. The randomization happened on a student-level within each class to ensure balance between the two conditions within each class. In the control condition, each student was assigned a random ranking of starred teachers.

This RCT was run on 2020 students (accounts) from 49 teachers who opted in to the TeacherASSIST Beta, from April 1, 2021 to April 7, 2021. I used their data from 1 week prior (March 25, 2021 to March 31, 2021) as their prior performance and 1 week after (April 8, 2021 to April 14, 2021) as their post performance. Standard TeacherASSIST without ranking was enabled during their prior performance week and post performance week. The post performance metrics of interest were 1) the percent of times the student answered the problem correctly without mistakes or additional supports (correct_first_try) 2) the percent of times the student requested additional supports e.g. hints, explanations, answer keys (use_help).

One major consideration of this RCT was that the difference between conditions come from two factors. The first factor was that the student support pool covered by the experimental condition was larger than the control. This was because, while the control only included student supports from starred teachers, the experimental condition included not only starred teachers but also other teachers who created more than 100 student supports within curricula and grades of interest, as previously specified. The second factor was the ranking itself. In order to identify the added effect of the ranking itself, the experimental condition could be compared to students in the control condition whose ranking was identical to that of the experimental condition barring the additional teachers introduced to the experimental condition.

7.2.3 Result

During the three weeks of the study, 77,650 problems were completed by 2,020 students across 49 teachers who opted into the ASSISTments TeacherASSIST beta. I found that, out of the 2,020 students, 642 did not work on any problems during the RCT week and 405 did not work on any problems during the post performance week (figure 7-3, 7-4, 7-5). In addition, 967 students never requested any student supports during the RCT week (figure 7-6). My hypothesis was that this was caused by irregularity in the contents taught in classes during the 3 weeks, that caused a massive number of students to complete no problems. This was most likely caused by Spring break or mid terms that was delayed due to COVID pandemic. This hypothesis was supported by how the retention of students was somewhat even across teachers as shown in figure 7-7.

In order to be able to reliably estimate the performance of students, I decided to filter out all students who completed fewerthan 3 problems during each of the prior performance, RCT, and post performance week. This was mostly caused by teachers not assigning any work to students through ASSISTments in said duration. In addition, I also filtered out students who requested for student supports fewerthan 3 times during the RCT week since those students did not experience the difference between conditions (student supports) enough. Only 188 students out of 2,020 students were left after the filter. I also removed all teachers who did not have at least one student left after the filter, which further removed 5 students from 188 students.

The filtered dataset contains 183 students, 99 of which were in the controlled condition and 84 were in the experimental condition. There was no significant difference between the percent of times students in the control and the experimental conditions requested for supports during the RCT week (control mean = 0.243, experimental mean = 0.238, p-value = 0.90). Unfortunately, this amount of data was insufficient to be used to investigate the effect of additional teachers alone after isolating the effect of ranking as students in the control condition were randomly assigned to random

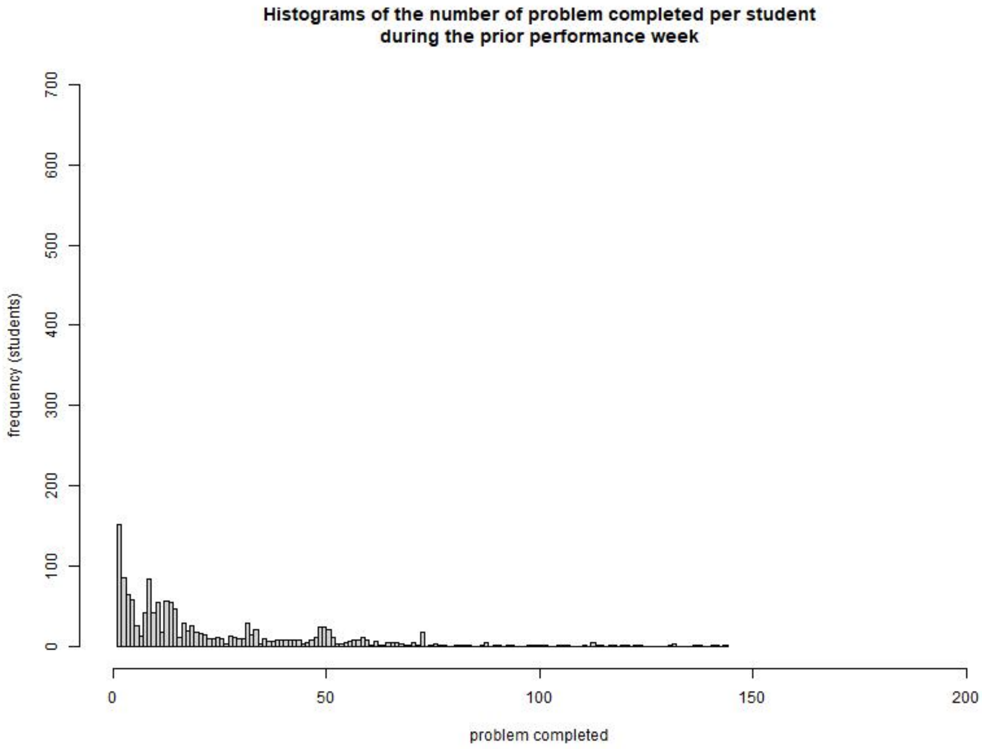


Figure 7-3: The distribution of the number of problems completed per student during the prior performance week.

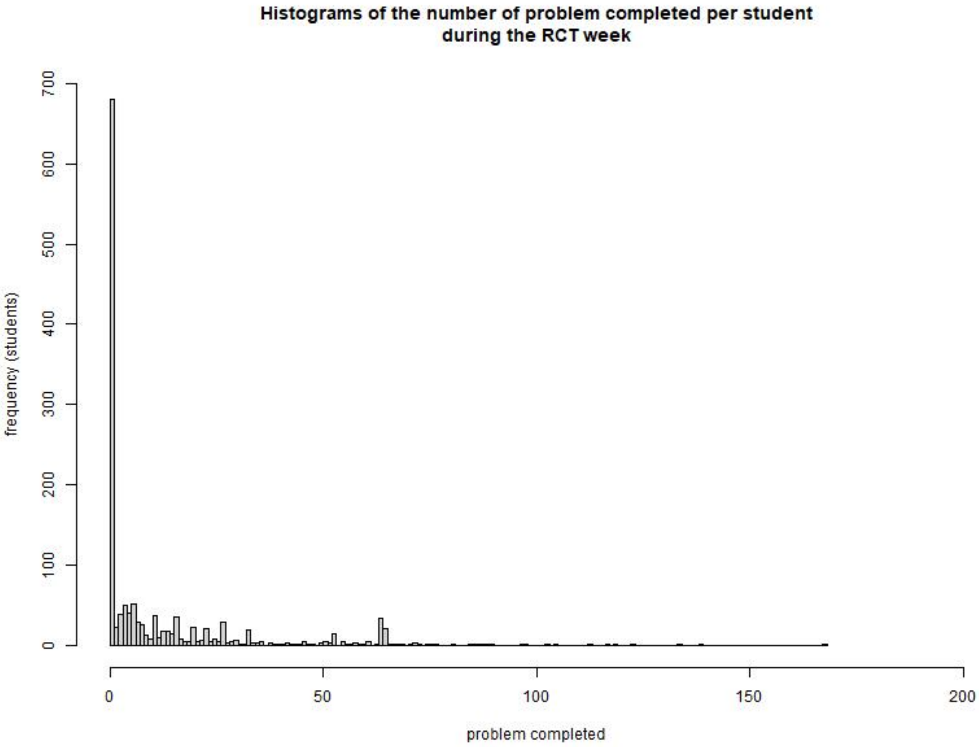


Figure 7-4: The distribution of the number of problems completed per student during the RCT week.

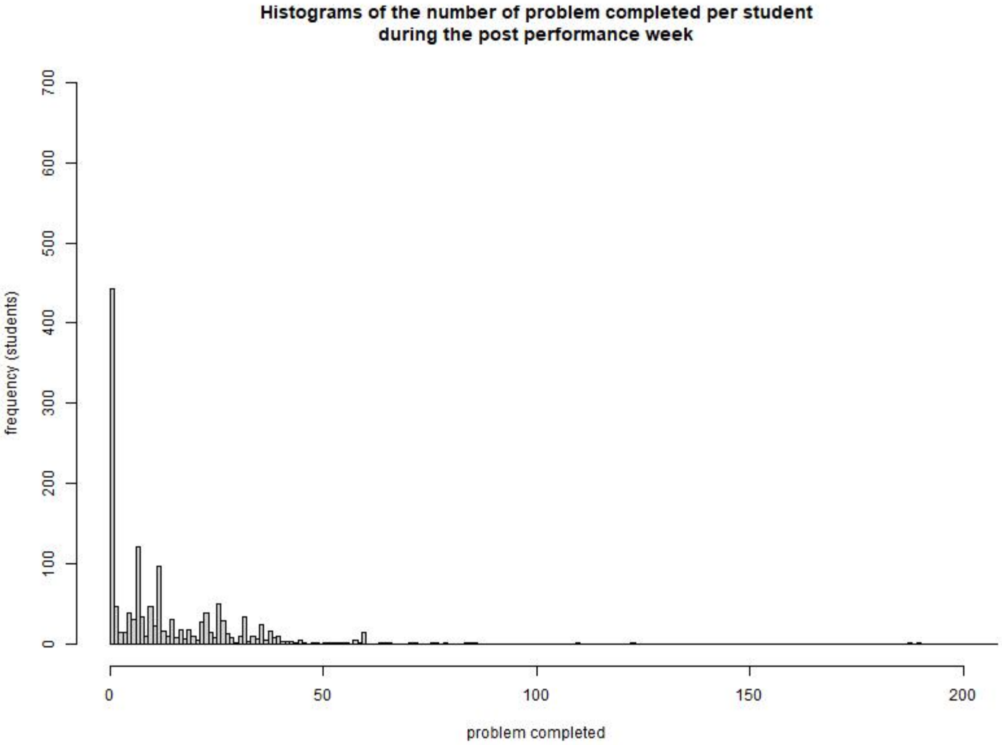


Figure 7-5: The distribution of the number of problems completed per student during the post performance week.

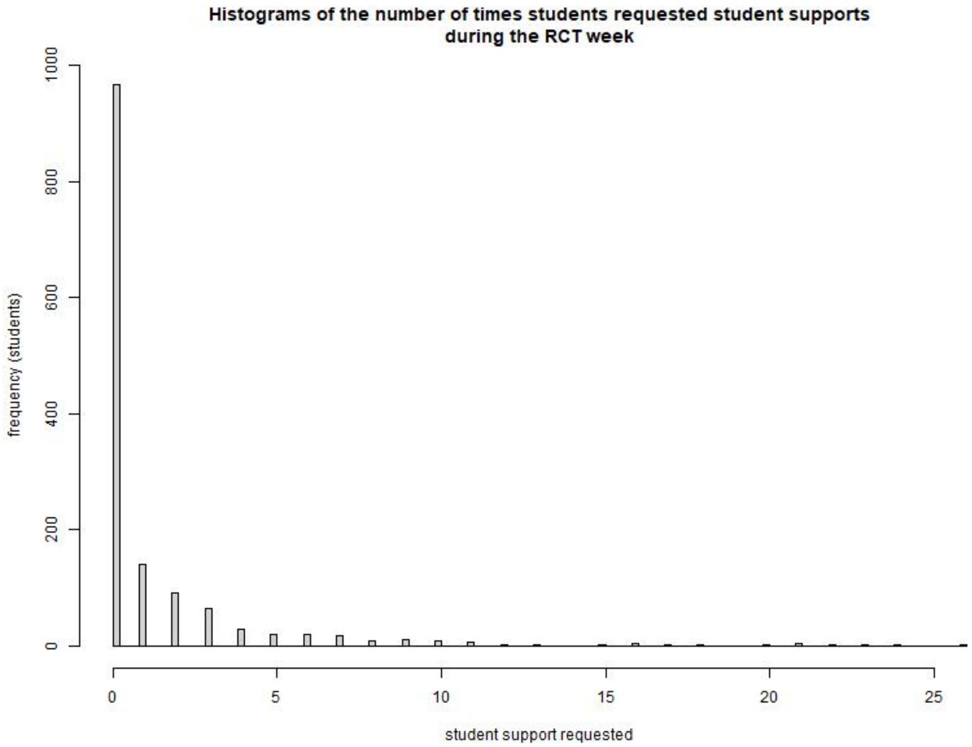


Figure 7-6: The distribution of the problems each student requested for student supports in total during the RCT week.

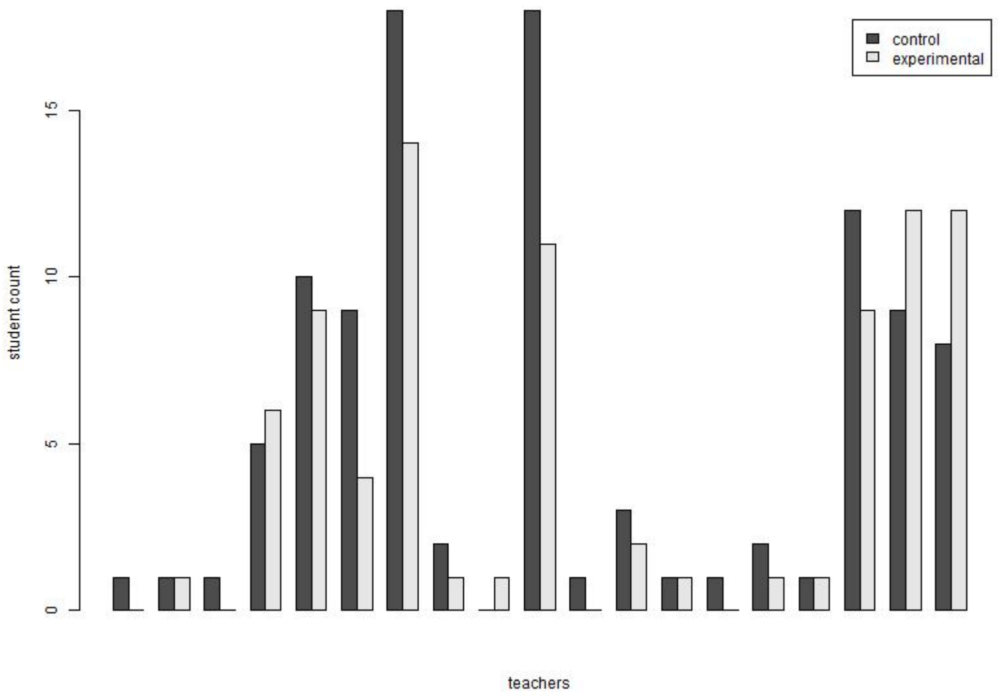


Figure 7-7: The distribution of the students left after the initial usage filter per teacher. Each pair of bars represents the number of students of those teachers whose usage exceeded the minimum usage threshold in the control condition and the experimental condition.

ranking of teachers.

Initially, I designed the RCT such that students were randomized within classes and teachers, which allowed me to use more complex models such as hierarchical linear models. However, the strange nature of data caused the number of students to drop to less than one-tenth of the original number, and the number of teachers from 49 to 14 (8 if I counted only teachers with more than 3 students in each condition). As a result, hierarchical linear models would be too complex and may give unreliable estimates.

Instead, I fitted linear regression models using student-level prior performance data to predict student post performance. The dependent measures of interest, i.e. student post performance, were 1) the percent of times the student answered the problem correctly without mistakes or additional supports (table 7.2) 2) the percent of time the student requested for additional supports e.g. hints, explanations, answer keys (table 7.3). Specifically, I used lm_robust from the R package called estimatr. lm_robust fitted a linear regression model using ordinary least squares with robust standard errors. In simple terms, robust standard errors are unbiased when the variability of a predictor is uneven across the range of values (heteroscedasticity).

In each model, I used the condition (control vs experimental), the percent of times students correctly answered problems without supports during the prior performance week (pre_correct_first_try_avg), the percent of times students requested supports during the prior performance week (pre_use_help_avg), the average number of attempts per problem during the prior performance week (pre_attempt_count_avg), and the average number of supports requested per problem during the prior performance week (pre_support_count_avg).

Table 7.2 shows the estimated regression coefficients of the regression model fitted to predict the percent of times the student answered the problem correctly without mistakes or additional supports. I found that the effect of condition was 0.032 (not statistically reliable, p = 0.106). Interestingly, the effect of pre_correct_first_try_avg was also not statistically reliable (0.115, p = 0.124). This was unexpected since prior student correctness was known to be the best predictor of future student correctness.

	Estimate	Std. Error	p-value	CI Lower	CI Upper
(Intercept)	0.324	0.081	<0.001	0.164	0.484
condition=experimental	0.032	0.020	0.106	-0.007	0.071
pre_correct_first_try_avg	0.115	0.074	0.124	-0.032	0.261
pre_use_help_avg	-0.307	0.137	0.026	-0.577	-0.038
pre_attempt_count_avg	-0.002	0.049	0.969	-0.099	0.095
pre_hint_count_avg	-0.001	0.001	0.059	-0.003	<0.001

Table 7.2: Estimated regression coefficients from a linear regression fitted to predict percent of times the student answered the problem correctly without mistakes or additional supports (correct_first_try) during the post performance week

	Estimate	Std. Error	p-value	CI Lower	CI Upper
(Intercept)	0.133	0.063	0.038	0.008	0.257
condition=experimental	-0.010	0.020	0.606	-0.049	0.029
pre_correct_first_try_avg	-0.111	0.065	0.088	-0.239	0.017
pre_use_help_avg	0.191	0.120	0.115	-0.047	0.428
pre_attempt_count_avg	0.040	0.039	0.306	-0.037	0.117
pre_hint_count_avg	-0.002	0.001	0.001	-0.003	-0.001

Table 7.3: Estimated regression coefficients from a linear regression fitted to predict percent of times the student request for additional supports e.g. hints, explanations, answer keys (use_help)

I hypothesize that this was an effect of the aforementioned anomaly in the dataset. While the effect size, at a glance, looked promising compared to pre_correct_first_try_avg, it was not statistically reliable and the experiment should be repeated to investigate the effect of distribution policy using global ranking.

Table 7.3 shows the estimated regression coefficient of the regression model fitted to predict the percent of times the student requested additional supports. Similarly, the effect of the condition was not statistically reliable (-0.010, p = 0.606). None of the prior performance statistics except for pre_hint_count_avg had significant effects on the likelihood that the students would ask for more supports. The confidence interval of the condition coefficient suggested that students who were in the experimental condition were between 3% more likely to 5% less likely to request student supports afterward. The change of this likelihood could be interpreted both positively and negatively, depending on the effect of the condition on other dependent measures.

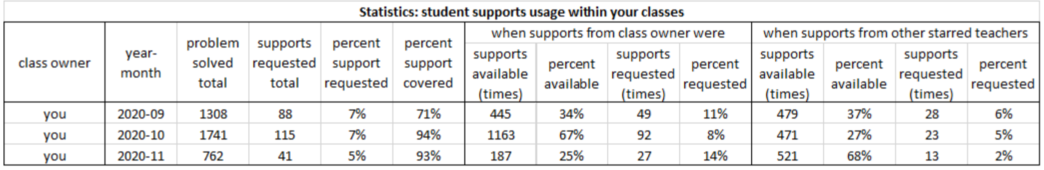
7.3 Feedback to Teachers who Created Student Supports

The last step of crowdsourcing is where contributors receive something back for their contribution: remuneration. While remuneration is in the form of monetary gains in many crowdsourcing tasks such as Amazon’s MTurk and YouTube, remuneration in crowdsourcing can also be something else as well depending on the contributors’ motivation to do the crowdsourcing task. For instance, StackOverflow’s remuneration is called r̈eputation point.C̈ontributors with high reputation points are given privileges based on set thresholds such as setting a ”bounty” on a question. A good implementation of this last step incentivizes users to keep contributing more to the crowdsourcing platform.

In this work, I chose to give ”remuneration” to teachers in the form of usage reports for 2 reasons. First, since the ASSISTments platform’s goal is to be free for teachers and students to use, monetary remuneration is not a sustainable solution. Reports, however, are easy to generate and maintain, and teachers inside ASSISTments are already used to seeing and using a variety of reports such as student assignment reports.

Second, the incentive for teachers to create student supports in the first place was to help their own students. In other words, they wanted what they created to be seen and used by students. Regular student assignment reports only show the teachers student support usages within a single assignment, but not the bigger picture. The aim of this first report was to allow teachers to see usages of student supports, theirs and otherwise, over each month. In addition, starred teachers currently have no way of knowing how student supports they created had been used outside of their classes.

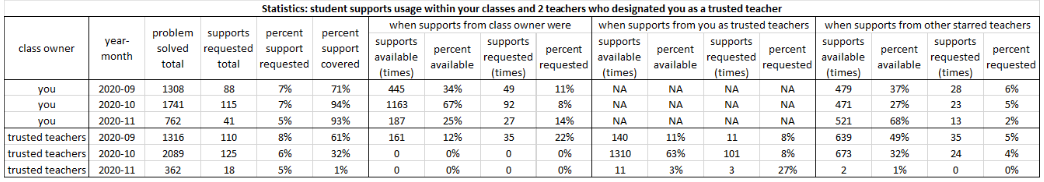
I designed 3 prototypes of reports based on different scopes of student support usages 1) report of student support usage within the teacher’s classes (figure 7-8 2) report of student support usage within the teacher’s classes and classes of teachers who have designated you as their trusted teacher (figure 7-9) 3) report of usage of student supports created by the teacher as a starred teacher (figure 7-10. The goal



Statistics: student supports usage within your classes

class owner	year-month	problem solved total	supports requested total	percent support requested	percent support covered	when supports from class owner were				when supports from other starred teachers			
						supports available (times)	percent available	supports requested (times)	percent requested	supports available (times)	percent available	supports requested (times)	percent requested
you	2020-09	1308	88	7%	71%	445	34%	49	11%	479	37%	28	6%
you	2020-10	1741	115	7%	94%	1163	67%	92	8%	471	27%	23	5%
you	2020-11	762	41	5%	93%	187	25%	27	14%	521	68%	13	2%

Figure 7-8: An example of a student support usage report within their own classes (first iteration).



Statistics: student supports usage within your classes and 2 teachers who designated you as a trusted teacher

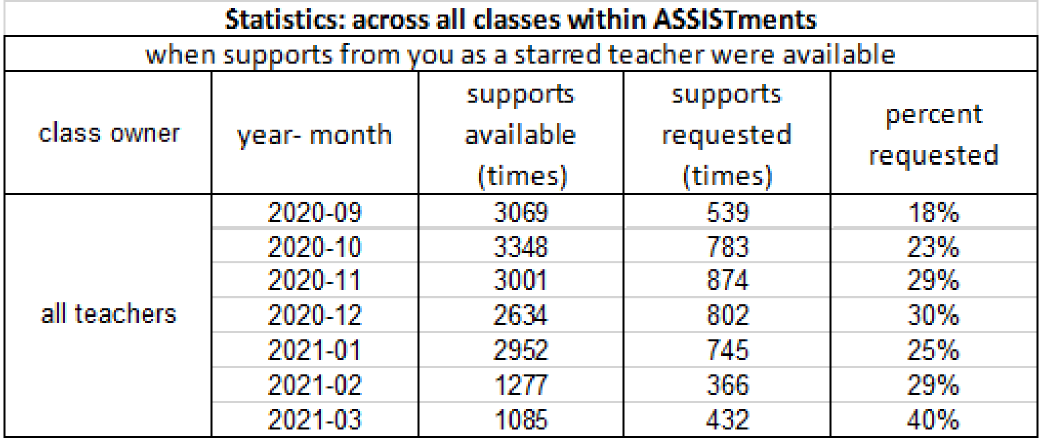
class owner	year-month	problem solved total	supports requested total	percent support requested	percent support covered	when supports from class owner were				when supports from you as trusted teachers				when supports from other starred teachers			
						supports available (times)	percent available	supports requested (times)	percent requested	supports available (times)	percent available	supports requested (times)	percent requested	supports available (times)	percent available	supports requested (times)	percent requested
you	2020-09	1308	88	7%	71%	445	34%	49	11%	NA	NA	NA	NA	479	37%	28	6%
you	2020-10	1741	115	7%	94%	1163	67%	92	8%	NA	NA	NA	NA	471	27%	23	5%
you	2020-11	762	41	5%	93%	187	25%	27	14%	NA	NA	NA	NA	521	68%	13	2%
trusted teachers	2020-09	1316	110	8%	61%	161	12%	35	22%	140	11%	11	8%	639	49%	35	5%
trusted teachers	2020-10	2089	125	6%	32%	0	0%	0	0%	1310	63%	101	8%	673	32%	24	4%
trusted teachers	2020-11	362	18	5%	1%	0	0%	0	0%	11	3%	3	27%	2	1%	0	0%

Figure 7-9: An example of a student support usage reports within their own classes with a trusted teacher (first iteration).

of these three prototypes was to allow teachers to see what kind of information could be available to them so that the reports could be iteratively improved upon.

The first prototype report was designed for all teachers who had created student supports for their students. The second prototype was a variation of the first prototype, with added information regarding usages of their student supports within classes of teachers who designated them as a trusted teacher as described in section 7.2. While, at the moment, the trusted feature was not live, the infrastructure was in place and ready to be used as soon as teachers designated other teachers as trusted. In fact, there's one teacher who was aware of this and informed that his two colleagues would like to designate him as a trusted teacher. The third prototype was designed for starred teachers so that they could see how much their student supports help other students.

I generated the report for 2 starred teachers and asked whether they found it useful. One of the teachers was unavailable due to personal reasons. The other teacher found the report prototypes to be overwhelming, contain too much information, and hard to digest. The only information he thought would be the most useful was how much his hints and explanations were used and how useful they were to the students, and would prefer to see the change week-by-week. In addition, the teacher was unfamiliar with the term "student supports" (parent category of hints and explanations)



Statistics: across all classes within ASSISTments

when supports from you as a starred teacher were available

class owner	year- month	supports available (times)	supports requested (times)	percent requested
all teachers	2020-09	3069	539	18%
	2020-10	3348	783	23%
	2020-11	3001	874	29%
	2020-12	2634	802	30%
	2021-01	2952	745	25%
	2021-02	1277	366	29%
	2021-03	1085	432	40%

Figure 7-10: An example of a usage report of student support they created outside their own classes (first iteration).

and preferred calling them hints and explanations. Thus, I designed a second set of report prototypes based on teacher feedback. Figure 7-11 and 7-12 showed the usage and helpfulness of student supports. For starred teachers, the same set of similar graphs could be used except the comparison would be between your hints and explanations vs other starred teachers' hints and explanations.

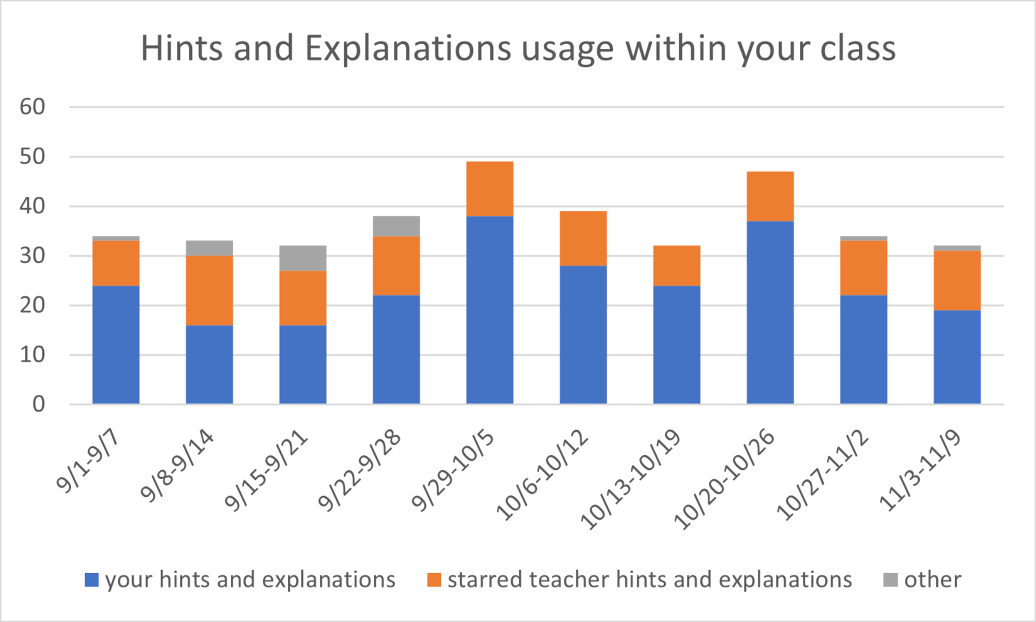


Figure 7-11: An example of a student support usage report within their own classes (second iteration).

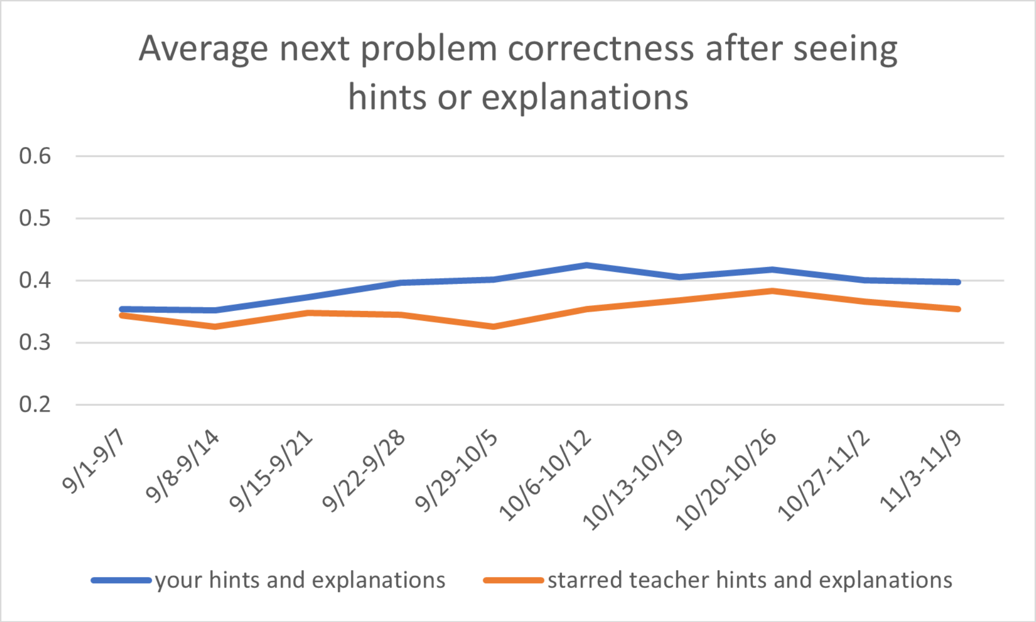


Figure 7-12: An example of a student support helpfulness report within their own classes (second iteration).

7.4 Conclusion

Crowdsourcing tasks could be separated into 4 stages: selection, accessibility, aggregation, and remuneration [18]. Without proper selection of contributors, the system risked getting malicious contributors. Without accessibility and aggregation, the pool of contribution would be no more than a simple accumulation of contribution. Without remuneration, contributors would feel unrewarded and no longer contribute.

In my previous work [36], I created a crowdsourcing system called TeacherASSIST inside the ASSISTments online learning platform. This feature allowed teachers to create hints and explanations, together called ”student supports,” to help their students as they work on classwork and homework. In addition, the system also distributed student supports created by a pre-selected group of approved teachers (”starred teachers”) in case there were no student supports created by the teachers. Through a randomized controlled trial and a repeated study, I found that TeacherASSIST improved student learning with statistical significance.

That work solely focused on the first and third stages of crowdsourcing: allowing teachers to create student supports for their students, and only allowing student supports by approved teachers to be distributed outside of their classes, with a random aggregation method. In this work, I improved TeacherASSIST crowdsourcing workflow by improving the second (see & rate), third (ranking), and fourth stages (report & feedback).

First, I created a proof-of-concept student support rating using the Qualtrics survey tool, which allowed teachers to see and rate the usefulness of sampled student supports from teachers who created sufficient amounts of student supports. The fact that I only received 27 responses, even with monetary incentive, supported my hypothesis that rating student supports was not a practical feature in itself. Then, I calculated the rating per user, and used it to create a ranking of teachers. Then, I ran a randomized controlled trial comparing this ranking (experimental) against a random ranking of starred teachers (controlled). After filtering out students who did not work on a minimum number of problems during the experimental weeks, there were 183 students, 99 of which were in the control condition. It's important to note that the effect of the experimental condition came from 2 factors: the additional non-starred teachers and a ranking based on teacher rating.

Initially, the plan was to also compare the experimental condition against students in control who were randomly assigned a ranking that mirrors that of the experimental condition without the additional teachers. However, due to an unforeseen anomaly in the dataset, the number of students was reduced from 2,020 initially to only 183. This, when this experiment is rerun, should be redesigned to ensure that the effect of added teachers and the ranking could be analyzed.

When I investigated the effect of the experimental condition on students' performance during the week after they receive the treatment, I found suggestive but not statistically reliable effect of the conditions on any of the dependent measures of interest. Interestingly, the effect of the prior correctness, which was known to be predictive of subsequent correctness, was also not statistically reliable. I hypothesized that this was caused by the strange nature of the experimental duration in combination with COVID pandemic, and that the effect of the condition was masked by this strange nature. To prove my hypothesis, a repeated experiment should be run for a longer duration to increase the opportunities that the students can work on problems and see student supports.

Lastly, I constructed prototypes for student support usages for teachers who created them. The reports were meant to inform teachers of the fruits of their labors, that the student supports they created had helped many of their students and, for starred teachers, other students inside ASSISTments. After consulting with the teacher, I found that the first set of prototypes had too much information and were not easy to consume. After taking his comment into consideration, I made a second set of report prototypes, containing mainly the information the teacher said he would like to see. It should be noted that this design process should be iterative and continuously improved upon as TeacherASSIST grows. What I constructed here wasn't meant to be the be-all-end-all report, but rather a starting point that allows teachers to see what kind of information they could see.

7.5 Limitation

The result of this work was restricted by several limitations. Due to the ongoing pandemic in the 2020-2021 academic year, most classrooms had gone either fully online or semi-online. For most teachers, this was the first time that they taught classes in such manners. Thus, it's unavoidable that the data collected in this work had anomaly. It's also possible that the anomaly was partially caused by other reasons such as school activities and midterm exams, as many classes became absent entirely.

For the student support rating, I massively overestimated teacher interest, and received only 27 responses. This also supported my hypothesis that rating student supports would be practically useless to most teachers since, even with monetary incentive, very few teachers were interested. However, it's undeniable that if TeacherASSIST knew how each teacher would rate each student support, TeacherASSIST could provide the most fit student support for each question and student. Practically, such rating data could be estimated using statistical and data mining techniques. Alternatively, TeacherASSIST could employ methods like multi-armed bandit algorithms that could intelligently calculate how good each student support is on the fly.

In addition to reports, there were also many other types of non-monetary remunerations that were used in crowdsourcing platforms. In an online learning platform such as ASSISTments, allowing teachers to have a teacher professional page, similar to the Wikipedia contributor page, could serve as one. In such a page, a teacher could display links to their school websites, folders containing teaching materials they created, and a button that would allow other teachers to designate the teacher as trusted. This would serve as a connection between ASSISTments and the teachers, as well as a gateway for teachers to connect to each other.

In addition, as teachers create more student supports and as their student supports help more students, teachers could gain contribution points that grant them privileges inside ASSISTments, similar to Stack Overflow. For example, it could allow them to create groups inside ASSISTments that would allow members to communicate and share materials. Teachers with enough points may also choose to promote themselves to starred teachers, which in turn improves scalability of TeacherASSIST.

Bibliography

[1] Deanne M Adams, Bruce M McLaren, Kelley Durkin, Richard E Mayer, Bethany Rittle-Johnson, Seiji Isotani, and Martin Van Velsen. Using erroneous examples to improve mathematics learning with a web-based tutoring system. *Computers in Human Behavior*, 36:401–411, 2014.

[2] ASSISTments.org. 05 rct designs (skill builders) - assistments advanced builder instructions site. `https://sites.google.com/site/assistmentsadvancedbuilder/05-rct-designs-skill-builders`.

[3] Stephen P Balfour. Assessing writing in moocs: Automated essay scoring and calibrated peer review™. *Research & Practice in Assessment*, 8:40–48, 2013.

[4] Tiffany Barnes and John Stamper. Toward automatic hint generation for logic proof tutoring using historical student data. In *International Conference on Intelligent Tutoring Systems*, pages 373–382. Springer, 2008.

[5] Tiffany Barnes and John Stamper. Automatic hint generation for logic proof tutoring using historical data. *Journal of Educational Technology & Society*, 13(1):3, 2010.

[6] Sameer Bhatnagar, Nathaniel Lasry, Michel Desmarais, and Elizabeth Charles. Dalite: Asynchronous peer instruction for moocs. In *European Conference on Technology Enhanced Learning*, pages 505–508. Springer, 2016.

[7] Alex J Bowers. Grades and graduation: A longitudinal risk perspective to iden-

tify student dropouts. *The Journal of Educational Research*, 103(3):191–207, 2010.

[8] Daren C Brabham. *Crowdsourcing*. Mit Press, 2013.

[9] Patricia A Carlson and Frederick C Berry. Calibrated peer review and assessing learning outcomes. In *Frontiers in education conference*, volume 2, pages F3E–1. Citeseer, 2003.

[10] Albert Corbett, Megan McLaughlin, and K Christine Scarpinatto. Modeling student knowledge: Cognitive tutors in high school and college. *User modeling and user-adapted interaction*, 10(2-3):81–108, 2000.

[11] Paul Denny, John Hamer, Andrew Luxton-Reilly, and Helen Purchase. Peerwise: students sharing their multiple choice questions. In *Proceedings of the fourth international workshop on computing education research*, pages 51–58, 2008.

[12] Shayan Doroudi, Ece Kamar, and Emma Brunskill. Not everyone can write great examples but great examples can come from anywhere. In *Seventh AAAI Conference on Human Computation and Crowdsourcing*. AAAI, 2019.

[13] edX.org. Ai grading. `https://edx-ora-2.readthedocs.io/en/latest/architecture/ai_grading.html`.

[14] Bart Epstein and Chris Rush. *Crowdsourcing Efficacy Research and Product Reviews*. EdTech Efficacy Research Academic Symposium, 2016.

[15] Tessa HS Eysink, Ton de Jong, Kirsten Berthold, Bas Kolloffel, Maria Opfermann, and Pieter Wouters. Learner performance in multimedia learning arrangements: An analysis across instructional approaches. 2009.

[16] Nancy Falchikov and Judy Goldfinch. Student peer assessment in higher education: A meta-analysis comparing peer and teacher marks. *Review of educational research*, 70(3):287–322, 2000.

[17] Mingyu Feng and Neil T Heffernan. Informing teachers live about student learning: Reporting in the assistment system. *Technology Instruction Cognition and Learning*, 3(1/2):63, 2006.

[18] Fernando J Garrigos-Simon, Yeamduan Narangajavana, and José Luis Galdón-Salvador. Crowdsourcing as a competitive advantage for new business models. In *Strategies in E-business*, pages 29–37. Springer, 2014.

[19] Neil T Heffernan and Cristina Lindquist Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[20] Seiji Isotani, Bruce M McLaren, and Max Altman. Towards intelligent tutoring with erroneous examples: A taxonomy of decimal misconceptions. In *International Conference on Intelligent Tutoring Systems*, pages 346–348. Springer, 2010.

[21] Kim Kelly. *A Set of Experiments Investigating Methods to Improve Student Learning Through Self-Regulated Learning.* PhD thesis, Worcester Polytechnic Institute, 2018.

[22] Avraham N Kluger and Angelo DeNisi. Feedback interventions: Toward the understanding of a double-edged sword. *Current directions in psychological science*, 7(3):67–72, 1998.

[23] Jared E Knowles. Of needles and haystacks: Building an accurate statewide dropout early warning system in wisconsin. *Journal of Educational Data Mining*, 7(3):18–67, 2015.

[24] Raymond W Kulhavy, Mary T White, Bruce W Topp, Ann L Chan, and James Adams. Feedback complexity and corrective efficiency. *Contemporary educational psychology*, 10(3):285–291, 1985.

[25] Chinmay Kulkarni and Scott R Klemmer. Learning design wisdom by augmenting physical studio critique with online self-assessment. Technical report, Citeseer, 2012.

[26] Mary Ellen Lepionka. *Writing and developing your college textbook: a comprehensive guide to textbook authorship and higher education publishing*. Atlantic Path Publishing, 2008.

[27] Heng Luo, Anthony Robinson, and Jae-Young Park. Peer grading in a mooc: Reliability, validity, and perceived effects. *Online Learning Journal*, 18(2), 2014.

[28] Mehak Maniktala, Christa Cody, Amy Isvik, Nicholas Lytle, Min Chi, and Tiffany Barnes. Extending the hint factory for the assistance dilemma: A novel, data-driven helpneed predictor for proactive problem-solving help. *arXiv preprint arXiv:2010.04124*, 2020.

[29] Bruce M McLaren, Tamara van Gog, Craig Ganoe, Michael Karabinos, and David Yaron. The efficiency of worked examples compared to erroneous examples, tutored problem solving, and problem solving in computer-based learning environments. *Computers in Human Behavior*, 55:87–99, 2016.

[30] P Mitros, Vikas Paruchuri, John Rogosic, and Diana Huang. An integrated framework for the grading of freeform responses. In *The Sixth Conference of MIT's Learning International Networks Consortium*, 2013.

[31] Jaclyn Ocumpaugh, Ryan Baker, Sujith Gowda, Neil Heffernan, and Cristina Heffernan. Population validity for educational data mining models: A case study in affect detection. *British Journal of Educational Technology*, 45(3):487–501, 2014.

[32] Jaclyn Ocumpaugh, Maria Ofelia San Pedro, Huei-yi Lai, Ryan S Baker, and Fred Borgen. Middle school engagement with mathematics software and later interest and self-efficacy for stem careers. *Journal of Science Education and Technology*, 25(6):877–887, 2016.

[33] Korinn Ostrow and Neil Heffernan. Testing the multimedia principle in the real world: a comparison of video vs. text feedback in authentic middle school math assignments. In *Educational Data Mining 2014*, 2014.

[34] Korinn S Ostrow and Neil T Heffernan. Studying learning at scale with the assistments testbed. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 333–334, 2016.

[35] Zachary A Pardos, Ryan SJD Baker, Maria OCZ San Pedro, Sujith M Gowda, and Supreeth M Gowda. Affective states and state tests: Investigating how affect and engagement during the school year predict end-of-year learning outcomes. *Journal of Learning Analytics*, 1(1):107–128, 2014.

[36] Thanaporn Patikorn and Neil T Heffernan. Effectiveness of crowd-sourcing on-demand assistance from teachers in online learning platforms. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pages 115–124, 2020.

[37] Thanaporn Patikorn and Neil T. Heffernan. Release of teacherassist dataset #1, 2020. Accessed: 2020-05-15.

[38] Drew Paulin and Caroline Haythornthwaite. Crowdsourcing the curriculum: Redefining e-learning practices through peer-generated approaches. *The Information Society*, 32(2):130–142, 2016.

[39] Maria Ofelia Pedro, Ryan Baker, Alex Bowers, and Neil Heffernan. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. In *Educational Data Mining 2013*, 2013.

[40] Andrea Prado Tuma, Sy Doan, Rebecca Ann Lawrence, Daniella Henry, Julia H Kaufman, Claude Messan Setodji, David Matthew Grant, and Christopher J Young. American instructional resources survey: 2019 technical documentation and survey results. 2020.

[41] Leena M Razzaq and Neil T Heffernan. To tutor or not to tutor: That is the question. In *AIED*, pages 457–464, 2009.

[42] David Reider, Kirk Knestis, and Joyce Malyn-Smith. Workforce education models for k-12 stem education programs: Reflections on, and implications for, the nsf itest program. *Journal of Science Education and Technology*, 25(6):847–858, 2016.

[43] Steven Ritter, John R Anderson, Kenneth R Koedinger, and Albert Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic bulletin & review*, 14(2):249–255, 2007.

[44] Jeremy Roschelle, Mingyu Feng, Robert F Murphy, and Craig A Mason. Online mathematics homework increases student achievement. *AERA Open*, 2(4):2332858416673968, 2016.

[45] Maria OZ San Pedro, Ryan S Baker, Neil T Heffernan, and Jaclyn L Ocumpaugh. Exploring college major choice and middle school student behavior, affect and learning: what happens to students who game the system? In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 36–40, 2015.

[46] Tim R Sass. Understanding the stem pipeline. working paper 125. *National Center for Analysis of Longitudinal Data in Education Research (CALDER)*, 2015.

[47] Barry Schwartz. The paradox of choice: Why more is less. Ecco New York, 2004.

[48] Douglas A Selent. Creating systems and applying large-scale methods to improve student remediation in online tutoring systems in real-time and at scale. 2017.

[49] Valerie J Shute. Focus on formative feedback. *Review of educational research*, 78(1):153–189, 2008.

[50] John Stamper, Michael Eagle, Tiffany Barnes, and Marvin Croy. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education*, 22(1-2):3–17, 2013.

[51] James Surowiecki. *The wisdom of crowds.* Anchor, 2005.

[52] Tim Vogelsang and Lara Ruppertz. On the validity of peer grading and a cloud teaching assistant system. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 41–50, 2015.

[53] Jacob Whitehill and Margo Seltzer. A crowdsourcing approach to collecting tutorial videos–toward personalized learning-at-scale. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 157–160. ACM, 2017.

[54] Wikipedia. Wikipedia:User access levels — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Wikipedia\%3AUser\%20access\%20levels&oldid=960722670`, 2020. [Online; accessed 11-June-2020].

[55] Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. Axis: Generating explanations at scale with learnersourcing and machine learning. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 379–388. ACM, 2016.

[56] David Wood, Jerome S Bruner, and Gail Ross. The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, 17(2):89–100, 1976.