Not Everyone Writes Good Examples But Good Examples Can Come From Anywhere *

Shayan Doroudi,[1, 2†] **Ece Kamar,**[3] **Emma Brunskill**[2]
[1]Carnegie Mellon University
[2]Stanford University
[3]Microsoft Research
doroudis@uci.edu, eckamar@microsoft.com, ebrun@cs.stanford.edu

Abstract

In many online environments, such as massive open online courses and crowdsourcing platforms, many people solve similar complex tasks. As a byproduct of solving these tasks, a pool of artifacts are created that may be able to help others perform better on similar tasks. In this paper, we explore whether work that is naturally done by crowdworkers can be used as examples to help future crowdworkers perform better on similar tasks. We explore this in the context of a product comparison review task, where workers must compare and contrast pairs of similar products. We first show that *randomly* presenting one or two peer-generated examples does not significantly improve performance on future tasks. In a second experiment, we show that presenting examples that are of sufficiently high quality leads to a statistically significant improvement in performance of future workers on a near transfer task. Moreover, our results suggest that even among high quality examples, there are differences in how effective the examples are, indicating that quality is not a perfect proxy for pedagogical value.

Introduction

In new online learning and work platforms that attract large numbers of people, many learners either individually or collectively make artifacts over the course of their interactions in the platform. These learner-generated artifacts can potentially be used to impact the learning opportunities of future learners, via learnersourcing (Kim 2015). For example, in massive open online courses, students create open-ended artifacts such as essays, computer programs, designs, and mathematical proofs. These artifacts are often presented to other learners in peer-evaluation exercises, and while the primary purpose of this is to scale grading (Piech et al. 2013), some instructors have treated evaluating peer work as an explicit learning opportunity (Devlin 2013). In Scratch, the popular online programming community and learning environment for kids, learners are encouraged to remix other kids' programs (Resnick et al. 2009), which has been shown to serve as a pathway for learning (Dasgupta et al. 2016). Finally, in crowdsourcing platforms, many crowdworkers do large numbers of tasks for requesters. While a lot of crowdsourcing tasks on websites like Amazon Mechanical Turk are microtasks, which are relatively easy, do not require creativity, and require little training, recent research has investigated crowdsourcing more complex work (Kittur et al. 2013; Doroudi et al. 2016; Streuer et al. 2017). For such tasks, workers might create complex artifacts (e.g., product reviews or website designs). As such, these artifacts could be presented to other workers as an inspiration or means of better understanding how to perform the task.

Thus, one way in which *naturally* generated learner-sourced artifacts can be used is to bootstrap the creation of low-cost curricula where we might not have the tools or time to create a high quality curriculum from scratch. Crowdsourcing is a particularly interesting domain to explore the effects of peer-generated artifacts for learning, because crowdsourcing and human computation tasks often come from ill-structured domains, where we do not have existing curricula to help teach workers. Moreover, the types of complex crowdsourcing tasks could be evolving over time as the future of work evolves and requesters have new needs. In this paper, we explore the efficacy of presenting work generated by crowdworkers as examples to help future crowdworkers perform better on similar tasks. Using peer-generated work for training has the benefit of potentially forming a self-sufficient pipeline for improving the quality of crowd work over time, without requiring requesters to invest into training.

In particular, we examine how to effectively use peer-generated examples in the context of a task where crowdworkers read two Amazon product pages and write a review that compares and contrasts the two products. We first ran an experiment to test the efficacy of various ways of presenting peer-generated artifacts (i.e., showing a single example, showing pairs of examples, and showing worker-generated guidelines) on future task performance. We found

*The title of this paper is an allusion to a quote from the movie *Ratatouille*: "In the past, I have made no secret of my disdain for Chef Gusteau's famous motto: 'Anyone can cook.' But I realize, only now do I truly understand what he meant. Not everyone can become a great artist, but a great artist can come from anywhere."

†Doroudi is now affiliated with the University of California, Irvine.

that *randomly* presenting peer-generated examples or guidelines did not lead to statistically significant higher performance than providing no training, presumably because the average quality of peer-generated examples was low. In our second experiment, we found that showing peer-generated examples of sufficiently high quality can lead to improved performance on future tasks.

Further, analysis of experimental results suggests that not all high quality examples are equal in terms of learning outcomes. Although future work is needed to characterize which characteristics of high quality examples correlate with better learning, our preliminary analysis indicates that simple surface-level features, like the formatting of an example, may be an indicator. These observations pave the way for future work on automatically searching the space of peer-generated examples to find ones that are more effective. By automatically discovering what makes a learnersourced example pedagogically effective, future work could not only serve the practical goal of bootstrapped example creation and consequently good learning outcomes for crowdsourcing workers, but also serve the scientific goal of determining what makes a good example good.

This paper makes several contributions to the crowdsourcing community. First, we demonstrate that for a product review task, showing random peer examples may not be an effective form of training crowdworkers. However, we find that showing two high quality examples can improve worker performance. Finally, we find that quality is not a perfect proxy for pedagogical effectiveness, and we find preliminary evidence that the formatting of peer examples correlates with learning outcomes. Moreover, this paper also contributes to the learning sciences by illustrating how learner contributions could benefit other learners, an idea which could also be of value to massive open online courses and even traditional classroom settings. The broader vision of this paper is envisioning how to create a self-sufficient pipeline to prepare workers for the ever-changing demands of the future work.

Related Work

Learnersourcing

The concept of *learnersourcing* refers to "crowdsourcing with learners as the crowd" often to help improve the educational experience of future learners (Kim 2015). Of most relevance to the present work are studies that have specifically looked at how to use learnersourcing to create new educational content that can help future learners (Williams et al. 2016; Heffernan et al. 2016; Whitehill and Seltzer 2017; Mitros 2015; Farasat et al. 2017; Glassman et al. 2016).

For example, Williams et al. (2016) developed a system called AXIS that had learners generate explanations to math word problems that could later be used to help other learners. They used a multi-armed bandit algorithm to automatically discover the explanations that learners found to be most useful. In a randomized experiment, they showed that learner-generated explanations that AXIS chose to present to students led to higher learning gains than explanations that did not meet a set of pre-specified quality checks. Their result is similar to ours in that it shows that not all learnersourced explanations are useful, but identifying good peer-generated explanations can be effective. However, they only compared their AXIS-chosen explanations to ones that were specifically thought to be bad, so it is not clear how effective random (or even above-median) explanations would have been. In our study, we show that random peer examples are not very effective, but high quality examples can be.

Moreover, AXIS and several other studies (Mitros 2015; Farasat et al. 2017; Glassman et al. 2016; Chiang, Kasunic, and Savage 2018) perform what is called *active learnersourcing* (Kim 2015), where learners are explicitly asked to do work that could be useful for future learners. For example, in the context of crowdsourcing, Chiang, Kasunic, and Savage (2018) developed Crowd Coach, a plugin where workers can give their peers short snippets of advice for how to do tasks while working on Amazon Mechanical Turk. In this paper, we are interested in *passive learnersourcing* (Kim 2015): leveraging artifacts generated from work that would be conducted regardless[1]. Passive learnersourcing has the advantage of not requiring additional work or cost to create curricula, which could be particularly useful in crowdsourcing settings where requesters have a limited budget.

There has been some prior work that could be considered passive learnersourcing in the context of training crowdworkers. For example, Zhu et al. (2014) compared having crowdworkers review other crowd work against simply doing more crowdsourcing tasks, and found that reviewing other workers' work was a more effective form of training. Similarly, in prior work, we compared validating other workers' solutions to complex web search tasks against using expert examples and doing more tasks (Doroudi et al. 2016). We found that validating other workers' solutions could be potentially as effective, and possibly more effective than, reading expert examples, *if* the solutions that were being validated were sufficiently long. This result resonates with our finding in this paper that reading random peer-generated examples may not be a very effective form of training, but reading high quality examples can be effective. However, these works differ from the current paper in that here we test presenting crowd work to workers as examples instead of validation tasks. Validation might be more effective in some cases, but it requires extra work before workers can tackle the tasks themselves.

Crowdsourcing as Learning at Scale

In addition to work on learnersourcing, there is an emerging body of work studying learning in crowdsourcing platforms. There are a number of studies that have looked into various ways of training crowdworkers, in addition to the ones mentioned above (Oleson et al. 2011; Dow et al. 2012; Singla et al. 2014; Mamykina et al. 2016; Streuer et al. 2017; Wang, Hicks, and Luther 2018). Beyond simply training crowdworkers to do better on particular tasks, recent work has looked into understanding how crowdsourcing platforms

[1]While we do consider using guidelines created by workers as a form of active learnersourcing, we primarily focus on directly using product reviews written by workers as examples.

can support learning as part of crowdwork and to foster the longer term development of worker skills (Krause et al. 2016; Dontcheva et al. 2014; Suzuki et al. 2016; Jun, Arian, and Reinecke 2018). For example, Jun, Arian, and Reinecke (2018) showed that workers value learning about scientific studies that they participate in. Our work fits into this narrative of crowdsourcing platforms as not just platforms to test learning at scale ideas, but to *enact and support* learning at scale.

Ill-Structured Domains

An ill-structured domain is a domain where the problem space is not (or cannot be) entirely well-specified and as such we do not have algorithms that can solve problems from such a domain (Newell 1969). According to Newell (1969), an ill-structured problem is one which humans can often solve but known algorithms (at least under the current state-of-the-art) cannot. It is thus natural that crowdsourcing tasks are often from ill-structured domains, because the very reason we resort to solving them with people is that we cannot do so with computers. As such, they can also be difficult to teach.

Prior work in educational psychology and the learning sciences has explored instruction for particular ill-structured domains. Some work has shown that examples can help novice learners on retention and *near transfer tasks* (i.e., tasks that share similar features with the example), but less so on *far transfer tasks* (i.e., tasks that are somewhat different from the example but may involve similar high-level approaches) (Kyun, Kalyuga, and Sweller 2013). Other work has suggested that direct instruction (including giving an example for a single task) is *in principle* not beneficial for ill-structured domains (Spiro and DeSchryver 2009). Instead, researchers have suggested using multiple examples from different tasks in ill-structured domains (Spiro et al. 1988), so learners can understand the variety of distinct cases that fall into that domain. This earlier work motivates the methods we test for training crowdworkers. However, the present work expands on this literature on ill-structured domains by exploring the use of *peer-generated* examples for teaching in ill-structured domains.

Task Design

The task we study here is writing product comparison reviews, which serves as an example of a subjective ill-structured task. Workers are given links to two Amazon pages for products that are somewhat similar but with several salient differences. Writing product reviews is a crowdsourcing task that has been used in prior research in training crowdworkers (Dow et al. 2012; Zhu et al. 2014; Streuer et al. 2017). Prior work had crowdworkers review products that they own, but this limits the ability to use crowdsourcing to review particular products. Therefore, we wanted workers to review specific products available on Amazon. We chose to have workers compare two products, instead of reviewing one, both because the comparative nature of the task would help ground the content of the review and because product comparisons are a common service that many websites provide. Workers were instructed as follows:

> Please look through the following two product pages and write a summary review that compares and contrasts the two products. Try to make the review as useful as possible for someone who wants to choose which product to purchase.

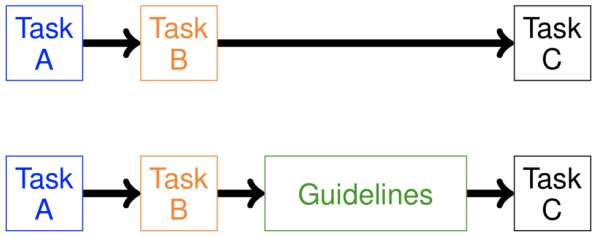
We had workers write reviews for three distinct categories of products: smoke alarms, board games, and gluten-free macaroni and cheese products. We choose product categories that are very different from one another so that we could test both whether it is useful to see examples of reviews for products from the same category (i.e., near transfer) as well as whether seeing examples from products of a different category can be useful (i.e., far transfer). We were particularly interested in identifying how to best support transfer, as one can imagine crowdsourcing requesters may need to have workers complete many similar tasks but for different categories, and the categories of interest might change over time. We used five tasks in particular: two for smoke alarms (Tasks A and A'), two for board games (Tasks B and B'), and one for macaroni and cheese (Task C). Tasks A and B were used to collect an initial pool of examples that we could show workers, and Tasks A', B', and C were used to test workers to see if examples improve their performance.

Experimental Design

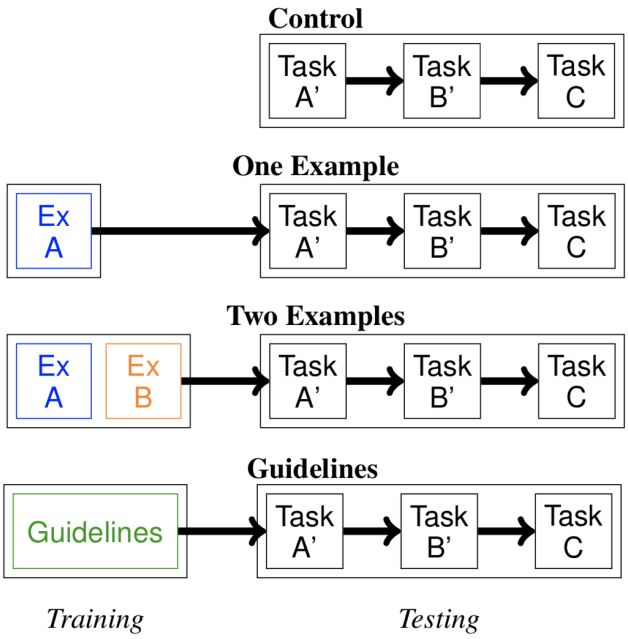
All of our experiments were conducted on Amazon Mechanical Turk, with the tasks being completed on Qualtrics. Workers who accepted our Mechanical Turk Human Intelligence Task (HIT) were first given a consent form and randomly assigned to one of several conditions (depending on the experiment). After giving consent, workers were given instructions possibly followed by some form of training (seeing examples or guidelines) depending on the condition they were assigned to. Workers were asked to spend as much time as they needed reading the examples or guidelines they were presented. They then completed up to three tasks in order (A', B', and C) followed by a short survey with qualitative questions about the difficulty of the task and usefulness of training. Workers were free to stop working on the tasks at any point in time, at which point they were given the survey. Workers were paid \$0.50 just for doing the HIT and survey, in addition to \$2 for each product comparison review they completed. They were told that they would receive the bonus payment provided that they follow the instructions. Additionally, workers were given \$0.50 for each example or guidelines that they had to read. The pay was chosen so that workers could expect a wage of \$11/hour[2] if they spent 10 minutes on each review, and as such workers were suggested to not spend more than 10 minutes writing each review.

After each experiment, we had workers grade the solutions both in terms of overall quality as well as checking whether the review contained specific features. We first released a HIT to test workers ability to accurately grade some gold standard reviews. Only workers who had participated

[2]This is in line with Dynamo's ethical standards for paying crowdworkers: http://wiki.wearedynamo.org/index.php/Fair_payment.



(a) Example Collection



(b) Experiment 1

Figure 1: Experimental design for (a) the example collection phase and (b) Experiment 1. The reviews and guidelines generated from the example collection phase are the examples and guidelines used in Experiment 1.

in the associated experiment were allowed to take the HIT. Workers who were qualified, were then given access to HITs for each review that needed to be graded. Each review was graded by three to five different workers using a rubric. The rubric consisted of two parts. First, the graders were asked whether or not the review mentioned particular points that we believed were worth mentioning in the product reviews (e.g., price and average star rating). Second, the graders were asked to rate the overall quality of the solution using the following scale:

5. It's hard to imagine a more useful resource for someone to decide which product to buy. The review appears to contain no factual errors.
4. The review would help you decide which product is best, but could have had some more information or could have been structured better.
3. The review would be helpful, but you would need to do more research to decide which product to buy.
2. The review has some distinctions between the two products, but you basically need to do your own research from scratch to decide which product to buy.
1. The review is misleading or does not really contain useful information (e.g., contains a major factual error that could result in purchasing the wrong product).

	Task A	Task B	Task C
Example Collection	0.86	0.86	0.90
	Task A'	Task B'	Task C
Experiment 1	0.74	0.75	0.67
Experiment 2	0.75	0.87	0.88

Table 1: Intraclass correlation coefficients, ICC(1, k), of overall quality ratings given to reviews from each experiment.

In this paper, we only use this overall quality scale to evaluate the efficacy of the different types of training. To measure the inter-rater reliability of the quality scores, we computed the intraclass correlation coefficient, namely ICC(1, k), which measures how likely it is that two randomly chosen samples of k graders would assign the same quality (1-5) for a given review (Shrout and Fleiss 1979). The intraclass correlation coefficients for all experiments are shown in Table 1. In the example collection phase, we used 5 workers to grade each review, while in the subsequent experiments, we only used 3 workers, which could explain why the correlation was a little lower for those experiments. However, overall the intraclass correlation tends to be high, meaning the average overall quality per review can be a reliable way of measuring review quality.

Example Collection

We recruited 70 Mechanical Turk workers to complete up to three tasks (A, B, and C). In addition to collecting examples from tasks, we also wanted to collect general guidelines for doing product comparison review tasks which we could also test. Workers were randomly assigned to one of two conditions. In one condition, workers just completed the three tasks, but in the second condition, after doing two tasks, workers were asked to write down general guidelines for doing the tasks. The experimental design is shown in Figure 1(a). We randomly assigned the workers to these two conditions in order to assess whether writing guidelines is beneficial to the workers who write the guidelines themselves, namely whether it improves their performance on Task C. 56 workers submitted acceptable work; work was rejected when workers clearly did not put an honest effort into the task, for example by copy-pasting text from the Amazon product pages (which the instructions explicitly told them not to do). This process resulted in 56 reviews for Task A, 47 reviews for Task B, and 41 reviews for Task C (which were not used as examples).

Condition	Num of Workers	Mean Overall Quality ± Std Dev		
		Task A'	Task B'	Task C
Control	92	2.43 ± 0.7	3.00 ± 0.7	*2.58 ± 0.7*
One Example	102	*2.27 ± 0.7*	*2.85 ± 0.7*	2.66 ± 0.7
Two Examples	97	**2.47 ± 0.7**	**3.02 ± 0.7**	2.67 ± 0.7
Guidelines	101	**2.47 ± 0.6**	2.94 ± 0.7	**2.68 ± 0.7**
One Example $\geq$	52	2.40 ± 0.7	2.87 ± 0.7	2.68 ± 0.8
One Example $<$	50	2.14 ± 0.7	2.83 ± 0.6	2.64 ± 0.7
Two Examples $\geq, \geq$	23	2.57 ± 0.8	**3.09 ± 0.8**	**2.86 ± 0.7**
Two Examples $\geq, <$	24	**2.65 ± 0.5**	3.01 ± 0.7	2.74 ± 0.7
Two Examples $<, \geq$	27	2.47 ± 0.7	3.07 ± 0.6	2.45 ± 0.6
Two Examples $<, <$	23	2.20 ± 0.6	2.90 ± 0.7	2.61 ± 0.8

Table 2: Experiment 1 Results. The highest-performing condition for each task is shown in bold and the lowest-performing condition is italicized. The highest-performing median-split condition for each task is also shown in bold. $\geq$ indicates above-median (greater than or equal to median) example quality and $<$ indicates below-median example quality.

There appeared to be no difference between the two conditions in the overall quality of Task C (2.4 for workers who created guidelines vs. 2.42 for workers in the control).

Experiment 1: Random Examples

In our first experiment, we wanted to test whether randomly presenting examples or guidelines improves the performance of workers on future tasks. In addition, we wanted to test what types of peer-generated artifacts are effective for training, both for near transfer tasks and far transfer tasks. We hypothesized that seeing examples help for near transfer tasks (e.g., seeing an example from Task A' would improve performance on Task A), while seeing a diversity of examples or general guidelines would lead to higher performance for far transfer tasks.

As such, we randomly assigned workers to one of four conditions. In the **control** condition, workers received no training. In the **one example** condition, workers saw a single randomly chosen example from Task A. In the **two examples** condition, workers saw a randomly chosen example from Task A followed by a randomly chosen example from Task B (both presented on the same page). Finally, in the **guidelines** condition, workers saw randomly chosen guidelines written by workers. These conditions are depicted in Figure 1(b)x. The workers were informed the examples and guidelines they saw were randomly chosen. Since randomly chosen examples could be of bad quality, workers were also provided the average overall quality score for each example that they saw and were told that both good and bad examples could help inform their work.

We recruited 433 participants from Mechanical Turk, of which 416 submitted acceptable work and 392 completed at least one of the test tasks.

Results

The results are shown in Table 2. We found no significant differences between the four conditions in terms of the average quality of the reviews for any of the three test tasks (Kruskal-Wallis test[3] with $p = 0.09$ for Task A' and larger p-values for the other two tasks).

Seeing two examples or guidelines generally resulted in performance that is comparable to or better than not receiving training across all tasks. However, seeing a single example appeared to be no better than the control, and trended worse on Tasks A' and Task B'. Interestingly, the biggest performance benefits (across all conditions) appear to be for Task C, which was designed as a far transfer task. This counters our hypothesis that diverse examples or guidelines might be needed for far transfer but not near transfer.

Post Hoc Analysis - Median Split Example Conditions We hypothesize that the reason none of these conditions appear effective—and that a single example might even be worse than no examples—is that these examples and guidelines were randomly chosen. To analyze this further, we looked at the results depending on whether the examples shown to workers were of above median or below median overall quality. The median example quality for Task A was 2.05 in the one example condition and 2.1 in the two examples condition and the median quality for Task B was 3.0 in the two examples condition. The results for these median-split conditions are shown at the bottom of Table 2. Seeing two examples of above median quality had the highest or second highest performance of all above/below median splits. However, interestingly, seeing a single example trended worse than the control on Task A' and B' regardless of whether the example was of above or below median quality.

Experiment 2: High Quality Examples

To test our hypothesis that seeing two high quality reviews may be pedagogically effective, we ran an experiment where

[3]The Kruskal-Wallis is a non-parametric hypothesis test, analogous to the ANOVA, that tests if there is a difference between the distributions for each condition.

Condition	Num of Workers	Mean Overall Quality ± Std Dev		
		Task A'	Task B'	Task C
Control	82	*2.23 ± 0.8*	*2.62 ± 0.8*	*2.57 ± 0.9*
Two Good Examples	69	**2.53 ± 0.7**	**2.72 ± 0.9**	**2.76 ± 1.0**
Example A1	18	2.26 ± 0.7	2.40 ± 0.8	2.58 ± 0.9
Example A2	27	2.60 ± 0.7	**2.83 ± 1.0**	2.81 ± 1.1
Example A3	24	**2.64 ± 0.7**	2.81 ± 0.7	**2.83 ± 0.9**
Example B1	20	2.45 ± 0.8	2.35 ± 0.9	2.67 ± 0.9
Example B2	20	**2.87 ± 0.7**	**3.04 ± 0.6**	**3.10 ± 0.8**
Example B3	29	2.34 ± 0.6	2.78 ± 0.9	2.6 ± 1.2

Table 3: Experiment 2 Results. The highest-performing condition for each task is shown in bold and the lowest-performing condition is italicized. The highest-performing example-split condition is also shown in bold for each task. Workers were free to leave the HIT at any time, so not all workers completed all the tasks.

Example	Mean Number of Newline Characters		
	Task A'	Task B'	Task C
Example A1	1.7	1.7	1.3
Example A2	2.3	2.2	2.0
Example A3	2.3	3.1	1.9
Example B1	*1.0*	*1.2*	*1.1*
Example B2	**3.0**	**3.6**	**2.3**
Example B3	2.4	2.5	2.0

Table 4: Average number of newline characters in reviews written by workers who saw each example. The example that resulted in the most newline characters for each task is shown in bold and the one that resulted in the least newline characters for each task is italicized.

we compared no training with showing workers two examples, drawn from one of the three highest overall quality score examples from each of Task A and Task B (**two good examples** condition). These examples are shown in Figures 2 and 3. Each example had much higher quality (between 3.6 and 4) than the median overall quality for its associated task. Other than the examples shown, everything in this experiment was the same as the previous experiment for the control and two examples conditions.

We recruited 161 workers on Mechanical Turk, of which 151 completed at least one task.

Results

The results are shown in Table 3. For each task, we ran a Mann-Whitney-U test[4] to see if the two good examples condition had significantly higher median overall quality than the control. The result was statistically significant for Task A' (p = 0.005) with a Glass' Δ effect size of 0.4, but not for the other two tasks. However, the trend seems to indicate that seeing two good examples was better than control for the other tasks as well.

[4]The Mann-Whitney-U test is a non-parametric hypothesis test, analogous to the t-test, that tests if two distributions are not equal.

Post Hoc Analysis - Specific Example Effectiveness A key motivation for the second experiment was that different examples may be of varying pedagogical value to future workers. The combined results of both experiments suggest that the quality of the example itself is an important feature, since high quality examples were associated with higher performance on subsequent tasks. However, it is natural to assume that there may be additional features that correlate with the teaching effectiveness of an example. In this section, we discuss our post hoc exploratory analysis towards identifying what makes an example pedagogically effective. To do so, we first examine if there are differences between the performance of workers who receive different pairs of high quality examples. We then examine how the formatting of the examples might explain some of these differences.

The idea that additional features may be important for pedagogical quality is supported by the bottom two sections of Table 3 which show the results for the two good examples condition subdivided by each particular example. Examples A2, A3, and B2 were associated with a positive effect on review quality for all tasks. Example A1 was generally no better than the control for all test tasks, and Examples B1 and B3 were no better than the control for some test tasks (and always worse than A2, A3, and B2). Moreover, although there were only a few workers who saw each specific pair

If you're looking for a hardwired solution, The First Alert is designed for that, plus it has a battery backup, in case you lose power. The First Alert is also the cheapest option.
The COOWOO Smoke Alarm is battery only, and while the manufacturer claims the battery will last 10 years, I seriously doubt that, and in general, you should replace your smoke alarm before then, anyways.
The First Alert, however, uses an ionization sensor, which tends to go off more frequently from cooking or other sources, and isn't recommended for more modern setups, while the COOWOO is photoelectric, which is considered more reliable, and has less of a chance of going off from someone burning their food on the stove.
I recommend the COOWOO for that purpose, despite it being slightly more expensive.

(a) Example A1

First of all, the price difference:
First Alert BRK 9120B Hardwired Smoke Alarm with Battery Backup - $12.56
10 Years Battery-Operated Smoke and Fire Alarm/Detector(Not Hardwired) with Silence Button and 10-Hours Eliminates Late Night Low Battery Chirps Mode Photoelectric Sensor & UL Listed Smoke&Fire Alarm - $$19.99

First Alert has a backup battery, so it will work in case of a power outage. It also has an ionization sensor, which detect smoke reliably. It can connect with other compatible alarms so it will all sound when one detects smoke. 10 year limited warranty.

10 Years Battery-Operated Smoke and Fire Alarm/Detector is not hard wired, so it will be easier to install. It has a long battery life of 10 years. 7 year warranty. The design of the case is more sleek and modern looking than First Alert.

(b) Example A2

COOWOO Smoke Alarm versus First Alert Smoke Alarm. One big difference in these two smoke alarms is that the COOWOO is not hard wired. Instead, the CooWoo is operated by a 10-year lithium battery whereas the First Alert is hard wired and relies on a 9-volt battery back up to keep your family safe during a power outage. The first alert also has an ionization sensor which helps to detect fast flaming fires.

The First Alert can be interconnected with up to 12 other compatible smoke alarms and six compatible devices such as repeaters, door closers, bells, and horns. If one unit triggers an alarm, all the smoke alarms will sound. There is also an indicator which will show which unit initiated the alarm.

On the other hand, the COOWOO has a photoelectric Sensor and an alarm sensitivity of 1.0 2.52%/ft. OBS. When the smoke alarm device detects particles of combustion and the concentration of smoke reaches the alarm threshold, the red LED flashes once per second and emits a loud pulsating alarm until the smoke is cleared. It has an alarm volume of >85dB(A) 3 meters.

(c) Example A3

Figure 2: Three examples for Task A (comparing First Alert and COOWOO smoke alarms) used in Experiment 2. The average quality of Examples A1, A2, and A3 were 3.9, 3.6, and 3.9 respectively.

of examples, our results suggest that some pairs of good examples might be especially effective. For example, the four workers who saw both Examples A2 and B2 had an average quality of 3.67 on Task C, which was substantially higher than the control (Glass' Δ effect size of 1.3) and the average for the two good examples condition.

However, recall that our measure of quality for each example was the average of three ratings by crowdworkers, which could be quite noisy. This could mean the examples that led to higher performance were actually of higher quality than examples that led to lower performance. To see if this was the case, we decided to get more accurate ratings for the quality of one example for each task that led to high performance (i.e., A3 and B2) and one example for each task that was seemingly ineffective (i.e., A1 and B3). We recruited nine crowdworkers who had not done any of our previous tasks to rate all four examples. The average ratings were 3.44 for A1, 3.67 for A3, 4 for B2, and 4.56 for B3. Based on these ratings, the two examples from Task A do not seem to differ substantially in quality, but B3 appears to be much better than B2. In particular, all nine workers rated B3 as being equal to or better than B2. However, B3 was the example which was expected to be least pedagogically effective. This again suggests that quality may not be a perfect proxy of pedagogical value.

Post Hoc Analysis - Effect of Formatting In looking at the actual examples (Figures 2 and 3), it appears that one important distinction between pedagogically effective and ineffective examples may be the example formatting, such as the number of newline characters. Indeed, examples that had more spacing (i.e., distinct paragraphs) tended to be better (with the exception of Example B3, which actually had the most spacing, but was not effective except on Task B').

In order to verify that the formatting actually has an effect on workers, we looked at the average number of newline characters in reviews written by workers who saw each example, as shown in Table 4. Workers who saw examples with fewer newline characters also on average wrote product comparison reviews with fewer newlines. A post hoc Mann-Whitney-U test shows that indeed workers who saw Example B2 appear to have used significantly more newline characters than workers who saw Example B1 on all three tasks ($p < 0.005$ on Task A' and B', and $p < 0.05$ on Task C, though we did not control for multiple comparisons).

Discussion

We have shown that seeing two high quality peer examples led to statistically significant higher performance on Task A' compared to not receiving training. The trends suggest that

Ticket to Ride tries to emulate cross country train journey in a board game while in Pandemic players work together to stop diseases from spreading. Both support similar number of players i.e. Pandemic 2-4 and Ticket to Ride 2-5. Play time is also similar with Pandemic advertising 45-60 and Ticket to Ride 30-60 minutes of game play. Both games are recommended for players aged 8 and above while the cost of Ticket to Ride is slightly more $49.99 than Pandemic $39.99. Main difference between the two games is obviously the game setting which is vastly different and hence game selection is mainly dependent on players preferences for that particular setting.

(a) Example B1

Pandemic and Ticket to Ride are both well reviewed board games worth considering. Pandemic is listed at $35.97 and is suitable for 2 - 4 players with a game running about 60 minutes. Ticket to Ride is priced at $44.97 and a game runs 30 - 60 minutes for 2 - 5 players.

Ticket to Ride is train adventure strategy game with the user traveling around the United States in a takeoff of "Around the World in 80 Days" with a winner takes all format. In Pandemic four diseases have broken out and players must work together as a team of specialists to save the world.

So, if you are in the mood to compete against other players, Ticket to Ride would be your choice. If you would prefer a cooperative game where all players win or lose together, Pandemic would be the better fit.

(b) Example B2

Today I am comparing two board games, Ticket to Ride and Pandemic, both which I personally own and love!

Both games will require at least 2 players to play! The nice thing about both of these games is that the game time is relatively fast and either game wont take more than 60 minutes to complete and both games are for ages 8+!

The main difference is that in Ticket to Ride you are playing alone, competing against everyone else to try and win, while in Pandemic you are working together to try and win the game! If you are trying to budget, Pandemic is also about $10 cheaper than Ticket to Ride.

Either way, you cannot go wrong as both games have over 4 stars and thousands of reviews!

(c) Example B3

Figure 3: Three examples for Task B (comparing the board games Ticket to Ride and Pandemic) used in Experiment 2. The average quality of Examples B1, B2, and B3 as rated by three workers were 3.8, 3.8, and 4 respectively.

two high quality peer examples also improved performance on the other test tasks. However, it could be that we did not have enough power to detect a significant effect on Tasks B' and C, both because there was a drop-out of workers after the first task and because the effect of examples might be smaller on Tasks B' and C. One reason for the examples having a smaller effect on the latter two tasks could be the fact that baseline performance on those tasks is higher than for Task A' (as seen in the control condition for both experiments as well as in the example collection phase). This could be because Tasks B' and C are easier and/or because worker performance improves as they do more tasks, which has been seen in prior work (Doroudi et al. 2016; Pan et al. 2016).

How Many Examples?

Although our first experiment suggested that a single example is not an effective form of training, we cannot say for sure if seeing one high quality example (e.g., Example B2) is not sufficient. On the other hand, if indeed seeing more examples is useful, especially seeing a diversity of examples to support generalization to different future tasks (Spiro et al. 1988), then perhaps seeing three or more examples could lead to higher performance than seeing two. However, we anticipate diminishing marginal returns and seeing many examples could lead to boredom, frustration, and/or lack of engagement with the examples, as we have seen in our prior work (Doroudi et al. 2016). Two examples seems to balance maximizing the effect of examples on future performance and avoiding worker frustration, but this should be confirmed in future work.

Multiple examples may be more effective if presented at the right time rather than all at once before workers perform the task. For example, Siangliulue et al. (2015) found in the context of an ideation task that giving examples of ideas is best when presented upon request or when individuals seem stuck, rather than presenting them in a regular interval. In our prior work, we found that providing an example solution for a task that workers have just completed as a form of feedback (e.g., as in a gold standard task), may actually be counter-productive (Doroudi et al. 2016). The optimal timing and number of examples will likely be task-dependent, and more investigation is needed to determine how to best present examples for different kinds of tasks.

Examples and Transfer

We initially hypothesized that seeing multiple examples or guidelines would be more beneficial for near transfer tasks; however, our results do not indicate this was the case. In Experiment 1, we found that performance increases (although not significant) were highest on Task C (far transfer), for all forms of training. In Experiment 2, performance increases on Task C appear comparable to, if not greater than, performance increases on Task B' (near transfer). This may be because for some reason there is a lower barrier to learning how to write a good review for Task C than for the other tasks, *even though* Task C is a different kind of task. It would be interesting to investigate this further in future work.

Going Beyond Quality

Our results seem to further suggest that if we had only shown a particular pair of examples to all workers (for instance, Examples A2 and B2), we may have seen a significant improvement on Tasks B' and C. These results suggest that prop-

erly chosen examples can be pedagogically valuable for both near and far transfer tasks. Future experiments are needed to confirm this.

This also suggests the potential benefit of using machine learning to try to automatically find the best example. One approach would be to use a multi-armed bandit algorithm to select the best example, perhaps after narrowing examples to ones that are of high quality, so that the algorithm would converge more quickly to finding a good example. This approach would be similar to the AXIS system (Williams et al. 2016). However, unlike AXIS, our results suggest that we may not simply want to find examples that are rated as being of high quality, but rather, examples that actually lead to higher performance gains for workers.

Another approach is to fit a model that predicts how pedagogically valuable each example is. Prior work has examined fitting models to characterize the pedagogical value of crowdsourced examples and explanations (Aleahmad, Aleven, and Kraut 2010; Mustafaraj et al. 2018), but they have not actually used such models to select examples. (Krause et al. 2017) fit a model using natural language features to predict the perceived helpfulness of crowdsourced feedback on design work. Various natural language features, such as the complexity of the feedback, whether justifications were provided, and the feedback sentiment, were found to all correlate with helpfulness. The authors used this model to develop a critique rubric for feedback providers, and showed that such a rubric can improve the quality of provided feedback.

Such a model could potentially generalize to predicting the pedagogical value of peer-generated examples that we have never tested on learners before. Additionally, such a model could predict what kinds of features make up a good peer-generated example, contributing to the learning sciences literature. The exact features that are most relevant to pedagogical value are likely to be task dependent; for example, a good example solution to a math problem may look different than a good product review. An automated approach could help identify the right features for each task. A key challenge to this would be identifying the features of the examples that the model would use. We have seen that structural features such as formatting may be indicative of example quality, which agrees with prior work showing that clearly delineating sub-goals improves the efficacy of worked examples (Eiriksdottir and Catrambone 2011). Our own prior work has also shown that the length of a reviewed peer-generated solution could be a good indicator of its pedagogical value (Doroudi et al. 2016). Similarly, Krause et al. (2016) found that among all features in their model for predicting feedback helpfulness, the feedback length was the most correlated with perceived helpfulness. Thus, it is possible that structural features of examples can be good proxies for pedagogical effectiveness, but perhaps richer features that are based on the semantics of the examples, such as some of the natural language features considered by Krause et al. (2016), could improve the accuracy of predictions.

Conclusion

Using artifacts created by workers to help teach other workers is a promising way to provide training in a scalable fashion. We have shown that presenting examples generated by crowdworkers can help other crowdworkers perform better on writing product comparison review tasks if those examples are of sufficiently high quality. We have also found preliminary evidence that quality alone is not a perfect predictor of pedagogical value, and other features, such as formatting, might be useful in that regards. Our work takes one step in the direction of envisioning how to create ecosystems where on-demand work and learning are integrated and support one another in a symbiotic fashion.

Acknowledgements

This research was supported in part by a Google grant and a Microsoft Research Faculty Fellowship.

References

Aleahmad, T.; Aleven, V.; and Kraut, R. 2010. Automatic rating of user-generated math solutions. In *Proceedings of the 3rd International Conference on Educational Data Mining*. International Educational Data Mining Society.

Chiang, C.-W.; Kasunic, A.; and Savage, S. 2018. Crowd coach: Peer coaching for crowd workers' skill growth. *Proceedings of the ACM on Human-Computer Interaction* 2(CSCW):37.

Dasgupta, S.; Hale, W.; Monroy-Hernández, A.; and Hill, B. M. 2016. Remixing as a pathway to computational thinking. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 1438–1449. ACM.

Devlin, K. 2013. Maththink mooc v4 - part 6. https://mooctalk.org/2013/12/23/maththink-mooc-part-6/.

Dontcheva, M.; Morris, R. R.; Brandt, J. R.; and Gerber, E. M. 2014. Combining crowdsourcing and learning to improve engagement and performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 3379–3388. ACM.

Doroudi, S.; Kamar, E.; Brunskill, E.; and Horvitz, E. 2016. Toward a learning science for complex crowdsourcing tasks. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2623–2634. ACM.

Dow, S.; Kulkarni, A.; Klemmer, S.; and Hartmann, B. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, 1013–1022. ACM.

Eiriksdottir, E., and Catrambone, R. 2011. Procedural instructions, principles, and examples: How to structure instructions for procedural tasks to enhance performance, learning, and transfer. *Human Factors* 53(6):749–770.

Farasat, A.; Nikolaev, A.; Miller, S.; and Gopalsamy, R. 2017. Crowdlearning: Towards collaborative problem-posing at scale. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, 221–224. ACM.

Glassman, E. L.; Lin, A.; Cai, C. J.; and Miller, R. C. 2016. Learnersourcing personalized hints. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 1626–1636. ACM.

Heffernan, N. T.; Ostrow, K. S.; Kelly, K.; Selent, D.; Van Inwegen, E. G.; Xiong, X.; and Williams, J. J. 2016. The future of adaptive learning: Does the crowd hold the key? *International Journal of Artificial Intelligence in Education* 26(2):615–644.

Jun, E.; Arian, M.; and Reinecke, K. 2018. The potential for scientific outreach and learning in mechanical turk experiments. In *Proceedings of the Fifth Annual ACM Conference on Learning@ Scale*, 3. ACM.

Kim, J. 2015. *Learnersourcing: improving learning with collective learner activity*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Kittur, A.; Nickerson, J. V.; Bernstein, M.; Gerber, E.; Shaw, A.; Zimmerman, J.; Lease, M.; and Horton, J. 2013. The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, 1301–1318. ACM.

Krause, M.; Hall, M.; Williams, J. J.; Paritosh, P.; Prip, J.; and Caton, S. 2016. Connecting online work and online education at scale. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 3536–3541. ACM.

Krause, M.; Garncarz, T.; Song, J.; Gerber, E. M.; Bailey, B. P.; and Dow, S. P. 2017. Critique style guide: Improving crowdsourced design feedback with a natural language model. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 4627–4639. ACM.

Kyun, S.; Kalyuga, S.; and Sweller, J. 2013. The effect of worked examples when learning to write essays in english literature. *The Journal of Experimental Education* 81(3):385–408.

Mamykina, L.; Smyth, T. N.; Dimond, J. P.; and Gajos, K. Z. 2016. Learning from the crowd: Observational learning in crowdsourcing communities. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2635–2644. ACM.

Mitros, P. 2015. Learnersourcing of complex assessments. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, 317–320. ACM.

Mustafaraj, E.; Umarova, K.; Turbak, F.; and Lee, S. 2018. Task-specific language modeling for selecting peer-written explanations. In *The Thirty-First International Flairs Conference*.

Newell, A. 1969. Heuristic programming: Ill-structured problems. *Progress in Operations Research III* 360–414.

Oleson, D.; Sorokin, A.; Laughlin, G. P.; Hester, V.; Le, J.; and Biewald, L. 2011. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Human computation* 11(11).

Pan, S.; Larson, K.; Bradshaw, J.; and Law, E. 2016. Dynamic task allocation algorithm for hiring workers that learn. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 3825–3831.

Piech, C.; Huang, J.; Chen, Z.; Do, C.; Ng, A.; and Koller, D. 2013. Tuned models of peer assessment in moocs. *arXiv preprint arXiv:1307.2579*.

Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; et al. 2009. Scratch: programming for all. *Communications of the ACM* 52(11):60–67.

Shrout, P. E., and Fleiss, J. L. 1979. Intraclass correlations: uses in assessing rater reliability. *Psychological Bulletin* 86(2):420.

Siangliulue, P.; Chan, J.; Gajos, K. Z.; and Dow, S. P. 2015. Providing timely examples improves the quantity and quality of generated ideas. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, 83–92. ACM.

Singla, A.; Bogunovic, I.; Bartók, G.; Karbasi, A.; and Krause, A. 2014. Near-optimally teaching the crowd to classify. In *Proceedings of the 31st International Conference on Machine Learning*, 154–162.

Spiro, R. J., and DeSchryver, M. 2009. Constructivism: When it's the wrong idea and when it's the only idea. In *Constructivist Instruction*. Routledge. 118–136.

Spiro, R. J.; Coulson, R. L.; Feltovich, P. J.; and Anderson, D. K. 1988. Cognitive flexibility theory: Advanced knowledge acquisition in ill-structured domains. Technical Report 441, Center for the Study of Reading, University of Illinois at Urbana-Champaign.

Streuer, M.; Krause, M.; Hall, M.; and Dow, S. 2017. On-the-job learning for micro-task workers. In *Human Computation 2017 Works-in-Progress*.

Suzuki, R.; Salehi, N.; Lam, M. S.; Marroquin, J. C.; and Bernstein, M. S. 2016. Atelier: Repurposing expert crowdsourcing tasks as micro-internships. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2645–2656. ACM.

Wang, N.-C.; Hicks, D.; and Luther, K. 2018. Exploring trade-offs between learning and productivity in crowdsourced history. *Proceedings of the ACM on Human-Computer Interaction* 2(CSCW):178.

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

Williams, J. J.; Kim, J.; Rafferty, A.; Maldonado, S.; Gajos, K. Z.; Lasecki, W. S.; and Heffernan, N. 2016. Axis: Generating explanations at scale with learnersourcing and machine learning. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, 379–388. ACM.

Zhu, H.; Dow, S. P.; Kraut, R. E.; and Kittur, A. 2014. Reviewing versus doing: Learning and performance in crowd assessment. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 1445–1455. ACM.