Toward a Learning Science for Complex Crowdsourcing Tasks

Shayan Doroudi[1], **Ece Kamar**[2], **Emma Brunskill**[1], **Eric Horvitz**[2]

[1]Carnegie Mellon University
Pittsburgh, PA
{shayand, ebrun}@cs.cmu.edu

[2]Microsoft Research
Redmond, WA
{eckamar, horvitz}@microsoft.com

ABSTRACT

We explore how crowdworkers can be trained to tackle complex crowdsourcing tasks. We are particularly interested in training novice workers to perform well on solving tasks in situations where the space of strategies for solving the task is large and workers need to discover and try different strategies to be successful. In a first experiment, we perform a comparison of five different training strategies. We show that for complex web search challenges providing expert examples is an effective form of training, surpassing other forms of training in nearly all measures of interest. However, such training relies on access to domain expertise, which may be expensive or lacking. Therefore, in a second experiment we study the feasibility of training workers in the absence of domain expertise. We show that having workers validate the work of their peer workers can work as well as having workers review expert examples if we use a rule to pre-filter the task solutions that they validate. The results suggest that crowdsourced solutions of peer workers may be harnessed in an automated training pipeline.

Author Keywords

crowdsourcing; worker training; worked examples; peer review; education; web search

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

To date, crowdsourcing has been focused largely on tasks that can be solved without special training or knowledge. However, many interesting tasks cannot be solved by unskilled crowdworkers. Examples of such tasks include using web search to answer complicated queries, designing an itinerary for someone going on a vacation [28], and condensing an academic article to an accessible summary for the general public [13]. One approach to crowdsourcing such tasks is to decompose them into smaller subtasks that are easier to solve [12, 4, 2, 13, 28]. However, such task decomposition requires the careful design and engineering of task-specific workflows. We investigate the less-studied case of crowdsourcing tasks that cannot be decomposed in a straightforward manner. Specifically, we consider the class of complex problem solving tasks that satisfy the following three properties: (1) there is a large space of potential strategies that workers can use to solve a task, (2) workers have the capacity to solve the task by discovering and trying different strategies, and yet (3) a significant proportion of unskilled workers are unable to solve it correctly on their first attempt. We address the prospect of extending the reach of crowd work to these tasks by exploring methods for training unskilled workers to perform better in this class of complex problem solving tasks.

Little is known about how to optimally train crowdworkers to perform complex tasks in a cost-effective way. Experts may be unavailable or unwilling to invest time into training crowdworkers and in many cases requesters themselves do not understand how to solve their complex tasks let alone how to train others to solve them. Furthermore, there may be a large continuum of possible strategies for solving these problems, with different strategies being optimal in different instances of the task. The strategies used to solve the task may also need to change over time (e.g, to detect web spam, workers need to adapt to adversarial shifts in spammer strategies over time). As such, it can be unwieldy, if not impossible, to write a comprehensive a set of standing instructions on how to approach these tasks.

We explore how to best train workers to solve complex tasks by performing comparative analyses of different training techniques on a complex web search task. Complex web search is an interesting domain for crowdsourcing because the desired answer cannot be captured by simply querying a search engine once. Instead, workers need to explore and aggregate multiple sources of information to reach an answer. Complex web search is a prototypical example of a task requiring workers to develop complex strategies and where we expect performance to be improved with training.

We test four methods for training workers: (1) learning by simply solving additional tasks (**solution** condition), (2) performing gold standard tasks where they see an expert solution after first trying the task (**gold standard** condition), (3) reviewing expert example solutions (**example** condition), and (4) validating solutions created by other workers (**validation** condition). We test these conditions along with a no-training **control** condition, and find that expert examples surpass other

CHI'16, May 07-12, 2016, San Jose, CA, USA
2016 ACM. ISBN 978-1-4503-3362-7/16/05...$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858268

forms of training, while minimizing costs of training such as dropouts, payments, and total time spent. However, as acquiring expert examples can be costly and require the engagement of domain experts, we sought to see if we could effectively train crowdworkers with solutions developed during crowdsourcing. We found that filtering solutions for training based on the length of the solutions can yield higher learning gains than provided by the example condition. The results highlight the feasibility of developing automated training pipelines for crowdsourcing complex tasks that run in the absence of domain expertise.

RELATED WORK

Crowd Training

Several prior studies explore the training of crowdworkers [21, 29, 23, 6]. Oleson et al. proposed the use of gold standards as a form of training on relatively simple microtasks, but their primary focus was on the use of gold standards for quality assurance rather than on the efficacy of the gold standards for training [21]. Willett et al. used examples for training workers and for calibrating their work to match the requesters' expectations on visualization microtasks and found that workers exposed to examples generated higher quality responses than workers who did not [27]. Similarly, Mitra et al. used examples followed by a qualification test and found that this training improved the quality of workers' data annotations [18]. Singla et al. use machine learning to optimize which training examples to show workers in simple classification tasks [23]. Moving beyond microtasks, Dontcheva et al. propose constructing platforms that integrate training and crowdsourcing in a photo editing environment [6]. The Duolingo system[1] similarly combines language learning and a crowdsourced translation service in a single platform. The construction of the two latter approaches require domain-specific knowledge and engineering and can be quite costly to build. Of most relevance to the our work, Zhu et al. compare two forms of training. They find that reviewing the work of other workers is a more effective form of training than doing more tasks [29]. Additionally, Dow et al. show that having workers self-assess their product reviews or having experts give feedback on their product reviews improves the quality of subsequent reviews [7].

Web Search

Turning to the web search task domain, much literature has also been devoted to the question of how to teach or train individuals to perform search. Several articles offer guidelines for teaching people how to perform web search in traditional classroom settings [8, 16, 5]. Walker and Engel suggest a form of instruction where students engage in search tasks, record their answers and thought processes, and are then given feedback as a class on the strategies used and how to best approach the tasks [26]. This is very similar to our gold standard intervention, but over a longer timescale and with an instructor providing more tailored feedback. Some researchers have also developed systems to help users improve their search skills. For example, Bateman et al. developed a search dashboard that allows users to reflect on their search behaviors and compare them to that of expert searchers, and found that users' behavior changed over time when using the dashboard to compare their behavior to that of experts, suggesting that viewing expert examples helps searchers [1]. Several controlled experiments have shown the efficacy of various web search training interventions. Lucas and Topi showed that training users in Boolean logic helps them achieve higher accuracy in their later searches [17]. Harvey et al. showed that showing crowdworkers query suggestions that were more effective than their own enabled them to generate higher quality queries of their own [9]. Finally, Moraveji et al. showed that providing task specific tips (e.g., to use specific advanced search features on Google) enabled workers to more efficiently complete web search tasks, and efficiency gains were maintained for similar tasks after a week [19]. However, these interventions are very specific to web search tasks, and in the last study specific interventions were tailored to individual tasks. Rather we propose forms of training that we hope can easily be adapted to other complex crowdsourcing tasks without extensive domain knowledge on behalf of the requester. Overall, these works highlight the positive effects of different training strategies on web search, which supports using this domain in our experiments.

Learning Sciences

To develop hypotheses about the effectiveness of different forms of training, we turn to the learning sciences literature, where instructional interventions have been more intensively studied than in the crowdsourcing community. *Worked examples*, or expert step-by-step solutions to a task, have been shown to be an effective form of teaching [25, 22]. Research has shown the presence of the *worked example effect*: reviewing examples is more effective than solving the tasks for learning, at least for novices [24]. While the *expertise reversal effect* claims that for more advanced students the opposite is true—solving problems is more effective than reviewing examples [11]—more recent work demonstrated that in a less-structured domain, the worked example effect holds for both novices and advanced students [20]. This finding may be relevant to crowdsourcing complex web search task and other complex problem solving tasks, as they are less-structured than problems in many typical educational settings. Additionally, learning sciences research has shown that novices learn more from their peers than from experts when being trained directly on the task they are tested on [10]. However, expert examples have been shown to be more effective than peer examples on transfer tasks—tasks that share some but not all properties of the examples [10, 3, 15]. As each of our complex web search queries can be quite different from one another, we expect our tasks to be in the transfer regime. Our experiments aim to explore how these results generalize to the crowdsourcing of complex tasks.

HYPOTHESES

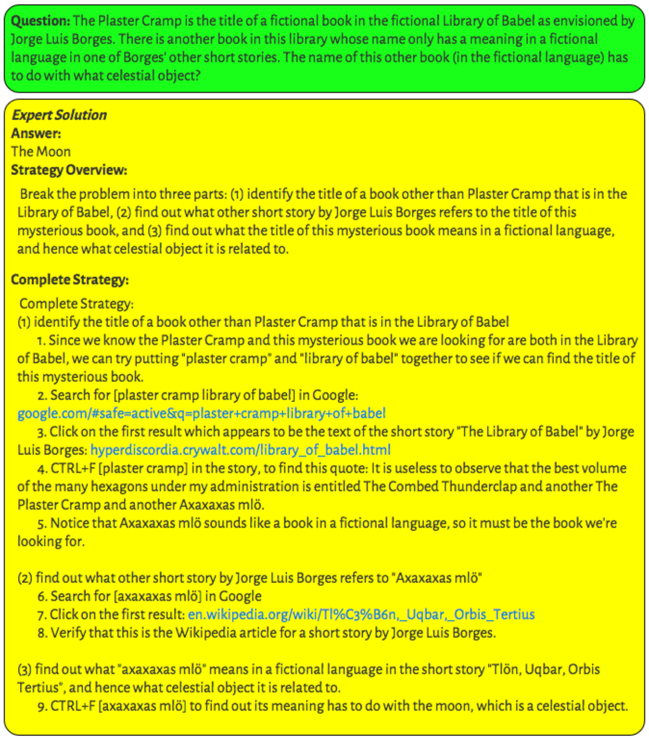
We formulated several hypotheses on the efficacy of various forms of training based on the prior findings in the literature. First, the worked example effect suggests the following hypothesis:

[1] www.duolingo.com

HYPOTHESIS 1. *Reviewing expert examples is an effective form of training in terms of increasing the accuracy of workers in finding answers to complex search tasks.*

Second, recall that Zhu et al. showed that reviewing the work of peer workers provides more effective form of training than doing more tasks [29]. This can be seen as an analogue to the worked example effect, but instead of simply reading through an example, the worker must review *and* validate the work of a peer worker. However, the learning science literature suggests expert examples are more effective than peer examples for transfer tasks [10, 3, 15]. These findings suggest the following hypothesis:

HYPOTHESIS 2. *Validating the work of others is also beneficial for increasing worker accuracy but less so than reviewing expert examples.*



Question: The Plaster Cramp is the title of a fictional book in the fictional Library of Babel as envisioned by Jorge Luis Borges. There is another book in this library whose name only has a meaning in a fictional language in one of Borges' other short stories. The name of this other book (in the fictional language) has to do with what celestial object?

Expert Solution
Answer:
The Moon
Strategy Overview:

Break the problem into three parts: (1) identify the title of a book other than Plaster Cramp that is in the Library of Babel, (2) find out what other short story by Jorge Luis Borges refers to the title of this mysterious book, and (3) find out what the title of this mysterious book means in a fictional language, and hence what celestial object it is related to.

Complete Strategy:

Complete Strategy:
(1) identify the title of a book other than Plaster Cramp that is in the Library of Babel
1. Since we know the Plaster Cramp and this mysterious book we are looking for are both in the Library of Babel, we can try putting "plaster cramp" and "library of babel" together to see if we can find the title of this mysterious book.
2. Search for [plaster cramp library of babel] in Google: google.com/#safe=active&q=plaster+cramp+library+of+babel
3. Click on the first result which appears to be the text of the short story "The Library of Babel" by Jorge Luis Borges: hyperdiscordia.crywalt.com/library_of_babel.html
4. CTRL+F [plaster cramp] in the story, to find this quote: It is useless to observe that the best volume of the many hexagons under my administration is entitled The Combed Thunderclap and another The Plaster Cramp and another Axaxaxas mlö.
5. Notice that Axaxaxas mlö sounds like a book in a fictional language, so it must be the book we're looking for.

(2) find out what other short story by Jorge Luis Borges refers to "Axaxaxas mlö"
6. Search for [axaxaxas mlö] in Google
7. Click on the first result: en.wikipedia.org/wiki/Tl%C3%B6n,_Uqbar,_Orbis_Tertius
8. Verify that this is the Wikipedia article for a short story by Jorge Luis Borges.

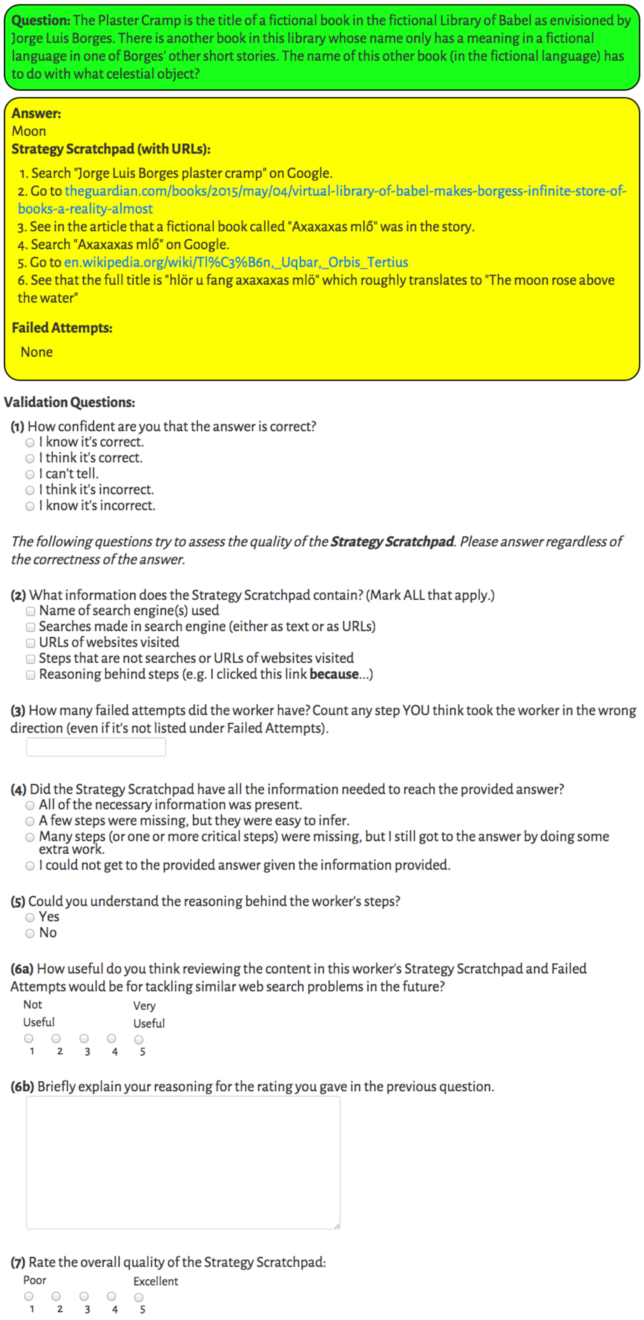
(3) find out what "axaxaxas mlö" means in a fictional language in the short story "Tlön, Uqbar, Orbis Tertius", and hence what celestial object it is related to.
9. CTRL+F [axaxaxas mlö] to find out its meaning has to do with the moon, which is a celestial object.

Figure 1. Expert example for training Question Y.

Similarly we hypothesize that validating higher-quality peer solutions, which are similar to expert solutions, will lead to more effective training than validating low-quality solutions. Furthermore, we might imagine that the validation process has a benefit beyond simply reading through an example, so the training benefit from validating such high quality peer solutions may even exceed that of reviewing expert examples. These hypotheses can be formulated as follows:

HYPOTHESIS 3. *Having workers validate filtered crowd-sourced solutions that are higher quality than average leads to a greater increase in accuracy than having them review unfiltered solutions.*

HYPOTHESIS 4. *If the solutions presented to workers are of high enough quality, this will have at least the same effect as presenting workers with expert examples.*



Question: The Plaster Cramp is the title of a fictional book in the fictional Library of Babel as envisioned by Jorge Luis Borges. There is another book in this library whose name only has a meaning in a fictional language in one of Borges' other short stories. The name of this other book (in the fictional language) has to do with what celestial object?

Answer:
Moon
Strategy Scratchpad (with URLs):

1. Search "Jorge Luis Borges plaster cramp" on Google.
2. Go to theguardian.com/books/2015/may/04/virtual-library-of-babel-makes-borgess-infinite-store-of-books-a-reality-almost
3. See in the article that a fictional book called "Axaxaxas mlő" was in the story.
4. Search "Axaxaxas mlő" on Google.
5. Go to en.wikipedia.org/wiki/Tl%C3%B6n,_Uqbar,_Orbis_Tertius
6. See that the full title is "hlör u fang axaxaxas mlö" which roughly translates to "The moon rose above the water"

Failed Attempts:

None

Validation Questions:

(1) How confident are you that the answer is correct?
○ I know it's correct.
○ I think it's correct.
○ I can't tell.
○ I think it's incorrect.
○ I know it's incorrect.

*The following questions try to assess the quality of the **Strategy Scratchpad**. Please answer regardless of the correctness of the answer.*

(2) What information does the Strategy Scratchpad contain? (Mark ALL that apply.)
□ Name of search engine(s) used
□ Searches made in search engine (either as text or as URLs)
□ URLs of websites visited
□ Steps that are not searches or URLs of websites visited
□ Reasoning behind steps (e.g. I clicked this link **because**...)

(3) How many failed attempts did the worker have? Count any step YOU think took the worker in the wrong direction (even if it's not listed under Failed Attempts).

(4) Did the Strategy Scratchpad have all the information needed to reach the provided answer?
○ All of the necessary information was present.
○ A few steps were missing, but they were easy to infer.
○ Many steps (or one or more critical steps) were missing, but I still got to the answer by doing some extra work.
○ I could not get to the provided answer given the information provided.

(5) Could you understand the reasoning behind the worker's steps?
○ Yes
○ No

(6a) How useful do you think reviewing the content in this worker's Strategy Scratchpad and Failed Attempts would be for tackling similar web search problems in the future?

Not Useful				Very Useful
○	○	○	○	○
1	2	3	4	5

(6b) Briefly explain your reasoning for the rating you gave in the previous question.

(7) Rate the overall quality of the Strategy Scratchpad:

Poor				Excellent
○	○	○	○	○
1	2	3	4	5

Figure 2. A validation task for training Question Y with a real worker solution.

Confirming these hypotheses would provide support for building domain-agnostic pipelines that train crowdworkers using their peers work. Such pipelines could improve the quality of training over time via methods for presenting the best peer solutions to workers. Eventually, such a pipeline could accrue a repository of high quality worked examples from crowd work *without requiring the requester to have domain knowledge*. Such a pipeline would have the additional benefit of providing quality control of work performed on complex tasks via peer validation.

TASK DESIGN

We test our hypotheses on a web search task where the goal is finding the correct answer of a complex web search query. Figure 1 shows one such query along with an expert solution. In addition to asking about the answer, workers are instructed to write down their thought process and to record each step (including all visited URLs) they take towards the answer in a web form that we call the **strategy scratchpad**. Workers are also asked to record unsuccessful strategies in what we call the failed attempts box. An example of a worker's solution is shown at the top of Figure 2. We found that workers were typically compliant with entering URLs, but many workers did not provide the rationale for the steps they took.

We developed a pool of complex web search questions that were designed to typically require several searches to find the right answer. Questions were adapted and influenced from search tasks given in "A Google A Day"[2] since these questions were found to be at the appropriate level of complexity. A sample question is shown in Figure 1. We ran a pilot study to decide how many questions to show in each training session. We hypothesized that using too many questions may decrease worker engagement with the study while using too few questions may decrease the effectiveness of training. After trying training sessions with 1, 2 or 3 questions, we found that some workers found it unreasonable to have to review three expert examples before being able to start the task. We settled on using two training tasks and five test tasks. We refer to the training questions in the rest of this paper as X and Y, and the test questions as A, B, C, D, and E. We note that optimizing the quantity of training is an interesting question that we do not explore further in this paper.

Experimental Design

We ran all of our experiments on Amazon Mechanical Turk.[3] Workers were assigned to one of several different training conditions (i.e. five in Experiment I and three in Experiment II) as soon as they accepted the task. The workers were assigned to the conditions in a round robin fashion to balance the number of workers assigned to each condition. Workers were first presented with an informed consent form that did not reveal we were studying worker training. Upon providing consent, workers were presented with condition specific instructions followed by two training tasks (unless they were in the control condition), possibly an additional set of instructions depending on the condition, and then five test tasks. For both training and test questions, we assigned the questions to workers in a random order. For example, workers were as likely to see training question X and then Y as they were to see Y and then X. While doing any of the tasks, the worker could choose to stop working on the task by completing an exit survey, which was required for payment. When workers began the survey, we revealed that the primary purpose of the study is to analyze the efficacy of various forms of training, and asked them several questions about the tasks in general and the efficacy of the training they received in particular.

EXPERIMENT I

The first experiment was performed to compare various forms of training inspired by the literature. We sought to find the most effective method for training as characterized by several metrics including worker accuracy. We focused on validating hypotheses 1 and 2, on exploring the relative efficacies of workers reviewing expert examples and validating peer-generated solutions.

Conditions

The five conditions we ran in the first experiment were as follows:

- **Control**: Workers receive no training. They are simply given instructions on how to perform the web search task and are then given the five test tasks (A, B, C, D, and E) in a random order.
- **Solution**: Workers are first presented with training tasks X and Y in random order as a form of training. Workers are given the same instructions as in the control condition, except that it tells them they will have seven tasks instead of five. They are not told that the first two tasks are for training. (We refer to this as the *solution* condition as workers are *solving* additional tasks for training.)
- **Gold Standard**: Workers start by solving two tasks for training as in the solution condition. However, after submitting the answer to each of these two tasks, workers are shown the correct answer to the task along with an expert solution. Workers are told in the initial instructions that they are going to receive the correct answers and expert solutions for the first two tasks but are also told that the expert solutions are more thorough than what we expect from them. Note that we do not use the phrase "gold standard" to refer to these tasks to workers since the term gold standard may have negative associations for workers in terms of disqualification or rejection of work. We refer to this condition as *gold standard* in the paper simply to refer to its analogy to gold standard tasks as used in the crowdsourcing literature [21].
- **Example**: Workers are given two expert examples for training. On the instructions given to workers for reviewing the examples, workers are informed that they cannot move on to the next task until 30 seconds elapses so that they are encouraged to spend time reading and understanding the examples. As in the gold standard condition, workers are also told that the examples will be more thorough than the task solutions we expect from them. We then provide explicit instructions for the web search task followed by the five test tasks. One of these examples is shown in Figure 1.
- **Validation**: Workers are first asked to validate two other workers' solutions for questions X and Y in a random order. The solutions to be validated are randomly chosen

[2] www.agoogleaday.com

[3] We used only workers from the United States who had at least a 98% approval rate.

Number of Workers (Percent of Workers that Start)				
	Start the HIT	Finish ≥ 1 training task	Finish ≥ 1 test task	Finish all tasks
Control	397	-	210 (0.53)	150 (0.38)
Solution	372	146 (0.39)	93 (0.25)	71 (0.19)
Gold Standard	372	142 (0.38)	95 (0.26)	72 (0.19)
Example	362	280 (0.77)	188 (0.52)	140 (0.39)
Validation	369	225 (0.61)	162 (0.44)	107 (0.29)

Table 1. Number of workers who start each condition and the retention rate at various points.

	Per Task (Pooled Across Test Questions A, B, C, D, E)			Per Worker		
	Accuracy	Time (min)	Strategy Length (char)	Accuracy	Total Time (min)	Training Payment
Control	0.48	8.28 ± 7.35	492 ± 385	0.50 ± 0.27	41.2 ± 22.2	$0.00
Solution	0.54	6.65 ± 6.33	477 ± 396	0.55 ± 0.28	55.2 ± 23.9	$2.43
Gold Standard	0.51	6.69 ± 4.47	467 ± 297	0.52 ± 0.21	54.7 ± 20.8	$2.44
Example	0.61	9.58 ± 7.15	625 ± 424	0.61 ± 0.26	49.6 ± 22.0	$0.20
Validation	0.55	9.47 ± 7.32	539 ± 339	0.56 ± 0.26	57.3 ± 24.6	$1.00

Table 2. Comparison across conditions on metrics of interest. Mean ± standard deviation is shown. Per task accuracy is a Bernoulli random variable, and hence as the accuracies are near 0.5, the standard deviation is nearly 0.5 for every condition. Per worker columns only include workers who do all five test tasks. The training payment column shows the amount we pay workers who do both training tasks; this is always the same for workers in the example and validation conditions (as long as they submit acceptable work), but for the solution and gold standard conditions, the average payment is shown.

from a pool of 28 solutions collected in a previous pilot study. An example of a validation task is shown in Figure 2. In each validation task, a worker sees the answer, strategy scratchpad, and failed attempts box of the solution they are validating, and are then asked a series of questions about the solution to be validated. Once they complete the two validation tasks they are given explicit instructions for completing web search tasks followed by the five test tasks.

We paid workers between $0.50 and $1.50 for completing a web search task (depending on whether or not they got the correct answer and completeness of the strategy), $0.50 for each validation task, and $0.10 for reviewing an expert example. Workers in the gold standard condition were only paid for solving the tasks and were not paid extra for reviewing examples, because we do not enforce them to read through the example. Additionally, we paid workers $0.50 for completing the survey. Workers who did not submit the survey were not paid at all, since their data could not be submitted to Mechanical Turk, which we made clear to workers.

Results

Quantitive Metrics

Table 1 shows how many workers were in each condition and the retention rates per condition: how many workers did at least one training task, how many did at least one test task, and how many did all of the tasks. We see that the control and example conditions had the highest retention rates at all points in time, and the solution and gold standard conditions have the least, with the validation condition in between. This is not surprising as the control condition has no training and the example condition offers the fastest form of training whereas the gold standard and solution conditions spend the longest time in the training phases. Workers may be more likely to drop out the longer they are in the task, and this could be due to either external factors that have nothing to do with the task or due to a variety of task-related factors such as boredom, annoyance with the task, the difficulty of the task, and/or the time spent appearing to be not worth the pay. All of these were expressed as reasons for dropping out in our survey. Nonetheless we find that even in the most time-consuming conditions (which took near an hour on average, but took up to two hours for some workers), nearly 20% of workers completed all tasks. Moreover, we find that in all conditions (except the control) around half of the workers who did at least one training task finished all of the tasks, suggesting that among workers who are willing to finish the first training task, there is roughly an equal proportion of highly committed workers in every condition.

Table 2 reports non-retention metrics for the various conditions. We are particularly interested in whether training increases the accuracy of workers on the test tasks, and, if yes, which forms of training are most effective at increasing worker accuracy. We report both the average per task (averaged over all test tasks) and the average accuracy per worker among workers who did all of five test questions. The average accuracy per worker is computed by first calculating the average accuracy for each worker on the five test questions they did, and then averaging this measure across the workers.[4]

We find that for both measures of worker accuracy, all training conditions outperformed the control condition of having

[4] The accuracy per worker for workers who did *at least one task* yields similar results. However, it is a more noisy measure since workers who did only one task have a much more noisy accuracy than workers who did all five, but in the aggregate average across workers, accuracy rates for workers who completed 5 tasks would be weighted equally with those that completed 1 task.

	Question A	Question B	Question C	Question D	Question E
Control	0.67	0.43	0.50	0.53	0.29
Solution	0.70	0.49	0.57	0.62	0.35
Gold Standard	**0.84**	0.26	0.62	0.59	0.25
Example	0.77	**0.50**	**0.72**	**0.65**	**0.42**
Validation	0.73	**0.50**	0.54	0.64	0.34

Table 3. Comparison across conditions of per task accuracy for each question. The condition with the highest accuracy for each question is bolded.

no training. The differences in per worker accuracy were significant based on the non-parametric Kruskal-Wallis test ($p = 0.0067 < 0.05$). Doing a post hoc analysis on the per worker accuracy using Mann-Whitney U tests, we find that the example condition was significantly better than the control after a Bonferroni correction for doing 4 tests. With a similar analysis on per task accuracy using two-proportion z-tests[5], we find that the example and validation conditions were significantly better than the control after a Bonferroni correction.

The example condition had the highest gains in accuracy over the control condition with an effect size of 0.25 (Cohen's h) (considered a small effect) for per task accuracy and 0.42 (Glass' Δ) (closer to a medium effect) for per worker accuracy. While these effect sizes are not considered large in the educational literature, we note that our form of training is *much* shorter than traditional educational interventions, so we do not expect effect sizes to compare to those of traditional interventions.

As for time spent per test task, we find that the example and validation conditions took longer than the control by over a minute on average, while the solution and gold standard conditions took less time than the control by over 1.5 minutes on average. Despite the large difference in time per task, we find that in total the example condition took less time on average for workers who did all of the tasks than the solution and gold standard conditions since the example condition spends much less time on training. Furthermore, the number of characters in the strategy scratchpad was greater for the example and validation conditions than the other conditions.

Finally, we do a comparison of the conditions on the per task accuracy for each of the five test questions, as reported in Table 3. We find that the example condition achieved the highest per task accuracy on all questions except for Question A, where the gold standard condition did much better than any other condition. On the other hand, we find that the gold standard condition did much poorer on Question B compared to all the other conditions. In the discussion section, we present a case study analyzing why the effectiveness of the gold standard condition may vary between tasks.

Qualitative Metrics

We are also interested in understanding workers' perceptions of the tasks and the training they received. Table 4 shows how effective workers thought the training they received was across various categories based on responses on a five-level Likert scale. Workers in the control condition were not asked these questions since they received no training. We find that the example condition had highest score in three of the four categories: efficacy in improving workers' understanding of the task, training workers to better describe their strategy, and training workers in finding the right answer. However, the solution condition had the highest score in being effective in improving workers' search ability in general, which is interesting as that is the condition where workers get the most practice doing searches on their own. Moreover, we find that in all conditions except for the solution condition, the scores in Table 4 monotonically decrease from left to right. That is, workers find the training to be most useful in understanding what we want of them (e.g. what to do, and how detailed to be in writing their strategy) and less useful in teaching them more generalizable strategies. Some workers made this explicit when asked to explain their answers to these survey questions (e.g. "The paid expert examples were VERY helpful in seeing *how you wanted my thought process structured.*" and "When I was doing the validation tasks, I felt like there was a lack of direction, leaving me in the dust for some parts of the task. With that in mind, when doing the web search tasks *I wanted to be as thorough as possible so that if someone had to validate my task, it would be simple and to the point.*").

Discussion

We found that the example condition outperformed the other conditions in overall per task accuracy and per worker accuracy (significantly outperforming the control) as well as in per task accuracy for all but one of the test questions. These results provide evidence for Hypothesis 1, that worked examples are an effective form of training. We also found that the validation condition had the second highest learning gains among all conditions, and that it had significantly larger per task accuracy than the control condition, partially confirming Hypothesis 2. As the difference between the example and validation conditions was not significant, we cannot definitively claim that validation tasks are less valuable for training than expert examples.

The benefit of the example condition is even greater when we consider that it minimized almost every cost of training. It was the least expensive form of training; we paid only $0.20 for training as opposed to $1.00-$3.00 for the other conditions. It also had the lowest dropout rates of any condition. One potential downside of using expert examples or validation tasks for training is that the average time per task is longer than for the other conditions. However, we saw that since reviewing the examples takes very little time as com-

[5] Not all of the assumptions of this statistical test are satisfied in our domain as answers for the same worker on different questions are dependent.

	Training was effective for...			
	Understanding Task	Describing Strategy	Finding Answers	Search Ability
Solution	3.53	3.55	3.43	**3.44**
Gold Standard	3.69	3.50	3.20	2.98
Example	**4.08**	**3.87**	**3.64**	3.25
Validation	3.96	3.72	3.32	3.08

Table 4. Perception of workers as to whether the training they received was effective on a number of different categories. Each question is a five-level Likert item, where 1 means the worker strongly thinks the training was not effective and 5 means the worker strongly thinks the training was effective in the category of interest. The condition that workers rated most highly for each category is bolded.

	Question A	Question B	Question C	Question D	Question E
Control	0.61	-	0.31	0.37	0.39
Solution	0.36	-	0.49	0.30	0.27
Gold Standard	**0.31**	-	**0.25**	**0.29**	**0.16**
Example	0.59	-	0.44	0.45	0.24
Validation	0.55	-	0.38	0.38	0.28

Table 5. Comparison of conditions on the proportion of wrong answers that are common intermediate answers. For each question, only one or two intermediate answers that are on the right path to the correct answer are considered. Nothing is reported for Question B because no such answers were identified for this question. The condition that achieves the smallest proportion of intermediate answers is bolded.

pared to doing solution tasks, the example condition actually took less time in total. Of course, this will depend on how many test tasks are given to workers and how fast workers get as a result of solving more tasks.

Although the average time spent per test task and average solution length are greater for the example and validation conditions than the others, we speculate this is for different reasons in the two conditions. Workers in the example condition see very long solutions (1271 and 1999 characters long), which are likely to promote them to write longer solutions than in other conditions. Workers in the validation condition get solutions to validate that are only 420 characters on average (since they were generated by peer workers) but are asked questions judging the quality of others' solutions. Thus, we hypothesize workers in the validation condition are writing longer solutions more so because of the questions that were asked of them in the validation task and possibly the process of having to validate the other worker's work, i.e. the validation task makes them realize how detailed they need to be in order for their own work to be validated properly.

While the example condition minimized nearly all the costs of training that we considered, there is still one hidden cost to the example condition, which is the cost of developing high quality expert examples. In general, developing expert examples may be time consuming for many complex crowdsourcing tasks, but more importantly, it requires expert knowledge of how to do the task, which a requester may not have. Since validation tasks do not have this hidden cost of training, it would be desirable to find a way to use validation tasks to outperform expert examples in training. We explore this in the second experiment.

Case Study: Gold Standard

Before moving to our second experiment, we reflect about the gold standard condition. One might assume that the gold standard condition should combine the benefits of the solution and example conditions, but that is not the case. We found that the gold standard condition performed worst among all training conditions per improving worker performance. Moreover, the gold standard condition led to far worse performance on Question B than in any of the other conditions including the control. However, we also saw that the gold standard condition far outperformed the other conditions in the per task accuracy of Question A. So what is the gold standard condition doing? While orthogonal to the main hypotheses of our study, we consider a case study to explore the effect of gold standard tasks in crowdsourcing. The case study provides an example of applying results from the learning sciences in crowdsourcing, and shows how the design of a training condition may have unexpected consequences.

We first describe a common type of wrong answer on the web search tasks. Most of the web search questions are best decomposed into more than one subtask, which must either be completed in series or in parallel to find the solution. Each of these subtasks themselves has an answer. A common mistake of workers is to provide an **intermediate answer** as the final answer. For example, in training question Y shown in Figure 1, we find that a common strategy (and the one used in the example) decomposes the task into three parallel subtasks, with the intermediate answers to the first two subtasks being "Axaxaxas mlö and "Tlön, Uqbar, Orbis Tertius." Indeed these are both common wrong answers to this question. Test questions A, C, D and E also have one or two common intermediate answers each. In Table 5, we show the proportion of wrong answers that are common intermediate answers on the path to the correct answer. We notice that the workers in the gold standard condition have the lowest likelihood of making this common mistake across all four questions. We hypothesize that attempting to solve a task first and then comparing their answer with the expert answer in the gold standard condition allows workers to pay attention to this common mistake type in the training phase, especially if they had made the mistake themselves, and prevent them from repeating this mistake on the test questions. Although the gold standard condition may be effective in fixing this particular mistake, it may not motivate workers to carefully study the provided solution in the way the example condition might do. As a result, workers in the gold standard condition may not learn new strategies to do the task, which might explain why this condition does not do so well overall.

This is analogous to students who are given solutions to a homework assignment only looking at what they did wrong, rather than reading through all of the solutions in detail. This hypothesis is supported by results reported in the learning sciences literature on feedback. Kluger and DeNisi suggest a theory of feedback interventions whereby "attention is limited and therefore only *feedback-standard gaps* that receive attention actively participate in behavior regulation" and feedback interventions "change the locus of attention and therefore affect behavior" [14]. This explains the difference between the effect of examples and gold standard conditions, and suggests that gold standard tasks may not have desired effects because of how students choose to engage with them. We speculate that asking workers to engage with gold standard tasks differently (e.g. asking them questions about how the expert approach compared to their own) would give benefits that are comparable to or better than simply using worked examples, which we leave as a direction to explore in future work.

With this background, we can better understand the specific behavior of the gold standard condition in our tasks. For Question A (where the gold standard condition was performing well), the average accuracy was quite high across all conditions meaning it was a relatively easy question. Indeed one of the only mistakes on that question was to give an intermediate answer. As the gold standard condition minimizes such answers, it does best on this question. On the other hand, Question B (where the gold standard condition was performing terribly) is the only question that does not have any common intermediate answers, so the gold standard condition does not have that benefit here.[6]

EXPERIMENT II

The results of Experiment I demonstrating the effectiveness of the example and validation conditions suggest that there might be hope for the validation condition to perform as well as the example condition if we only present workers with the "best solutions" to validate. Thus in this experiment we explore our two remaining hypotheses: that validating filtered solutions that are higher quality than average leads to a higher increase in accuracy than validating unfiltered solutions (Hypothesis 3), and that if these solutions are high enough in quality, validating them will be at least as effective as reading through expert examples (Hypothesis 4). Before we describe the experimental design, we first describe how we went about filtering solutions that we believed would be effective for the validation tasks.

Filtering Validation Tasks

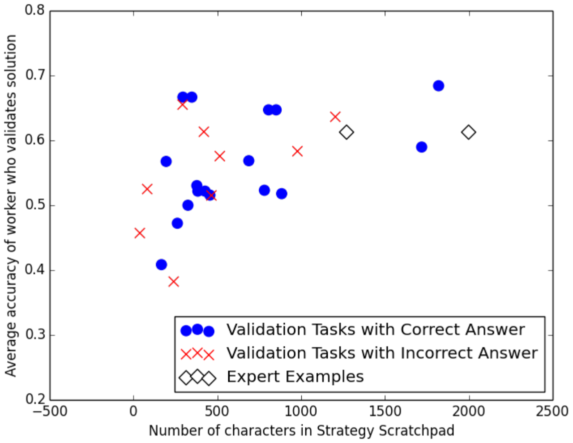


Figure 3. Average per worker accuracy on tasks done after seeing a particular validation task for training vs. the number of characters in the strategy scratchpad for that validation task. Each point represents a particular solution given as a validation task. The blue circles show solutions that arrived at the correct answer and the red x's show solutions that arrived at the wrong answer. The diamonds indicate the two expert solutions provided in the example condition for comparison; the average accuracy in this case is for all workers in the example condition.

We seek to answer the question "what properties of a solution makes it beneficial for training when presented as a validation task?" To help answer this question, we performed linear regression on a set of features for each of the solutions that was validated in Experiment I[7] to see how well they predict the per task accuracy of workers who saw the particular validation. The features for each validated solution include the answers provided for each quantitative question asked in the validation task (see Figure 2) averaged over workers who validated that solution. To this set of features we also added the number of characters in the strategy scratchpad for that task, the number of characters in the failed attempts box for that task, and the amount of time the original worker spent solving that task. We performed regularized linear regression (LASSO with a regularization parameter that was chosen using Leave-One-Out cross-validation). The resulting analysis

[6] This does not explain why it does poorer than all other conditions (including the control) on this question. One hypothesis, which is consistent with the theory on feedback, is that workers who get the training questions correct get a signal that they are doing well enough and therefore put less effort into subsequent tasks. In particular, in Question B, workers are told that the answer is a three-word name of a freeway. There are many such names that the worker can encounter while looking for the answer, and so perhaps workers in the gold standard condition give the first reasonable answer they find without verifying its correctness.

[7] We removed one one of the solutions that was a clear outlier. It had the longest solution the workers who validated it had a lower average accuracy than workers who validated any other solution, which violates the trend we discuss below. In addition to being a bad solution, it was formatted very strangely (without newline characters) and its length was due to long URLs; this seems to have had a negative effect on workers.

indicated that only the number of characters in the strategy scratchpad was correlated with accuracy[8].

Figure 3 shows for each solution presented as a validation task, the per worker accuracy (in the testing phase) of workers who validated that solution vs. the number of characters in the strategy scratchpad for that solution. The Pearson correlation coefficient is 0.46. We also see from the plot that whether the validation task had a correct or incorrect answer does not seem clearly correlated with the later accuracy of workers who validated it. This suggests that in this setting, regardless of solution correctness, longer solutions are generally more effective for training. Thus a requester could potentially decide whether a solution should be given as training as soon as the solution is generated by checking how long it is, without needing to first assess if the solution is correct.

Since our goal was to mimic the training process followed in Experiment I, in which all training conditions involved two tasks, our next task was devising a method for automatically identifying good pairs of validation tasks to present workers. We split validation tasks into "short" and "long" tasks by whether the solution length was longer or shorter than a single handset threshold. When we analyzed the effect of the different orderings of short and long solutions on worker accuracy on the data collected from Experiment I, we found that presenting a short solution followed by a long solution appears better than the other combinations for various thresholds. We note that we had very little data to evaluate presenting two long solutions. Choosing to present a short solution followed by a long one also has the practical advantage that all solutions collected from prior workers can be validated, which leads to automated quality control for all solutions collected from crowdworkers. In our second experiment, we test the efficacy of this approach for filtering solutions to be validated.

Experimental Design

Experiment II compared three conditions: **example-II**, **validation-II**, and **filtered validation**. Example-II and validation-II are the same as the corresponding conditions from the first experiment with a new worker pool. To see how the trends from Experiment I generalize when a new set of solutions is provided for validation, we refreshed the solution set for validation-II with solutions collected from Experiment I. The set included 100 solutions randomly sampled from those collected from the solution condition of Experiment I for questions X and Y as well as the 28 solutions used in the validation condition of the previous study.

The solutions used in the filtered validation condition came from the same randomly sampled set of 100 solutions generated in Experiment I. As before, the ordering of questions X and Y was randomized. The first solution workers validated was restricted to those shorter than 800 characters and the second solution they validated was always longer than 800 characters. This threshold of 800 characters resulted in 76 short and 24 long solutions used in the filtered validation condition.

Results

Table 6 displays how many workers were in each condition and the retention rates in each condition. Although our main focus is on how conditions compared within Experiment II, we note that the example-II condition had a lower retention rate than the earlier example condition, indicating the worker pool may have slightly changed. The validation-II and filtered validation conditions have similar retention rates.

Table 7 presents non-retention metrics. The example-II and filtered validation conditions had nearly identical performance on per task and per worker accuracy. These conditions perform slightly better than the validation-II condition, but the differences are not significant. Interestingly, there may be a regression to the mean effect between the first and second experiment, as the difference between standard validation and Example in Experiment I was larger (0.06 for worker accuracy), whereas here it is only a small gap (0.02).

Both the average time spent per test task and the average length of the strategy scratchpad on test tasks in the filtered validation condition is larger than in the other two conditions. This is likely because both the length of the solutions that workers see and the validation process itself affect the amount of effort workers put in subsequent tasks.

In Experiment I, we had a limited number of longer task length solutions provided to workers to validate, thereby limiting our ability to explore the effects of providing workers with two longer tasks to validate. However, a number of the solutions presented to workers in Experiment II (i.e. solutions generated during Experiment I) had a longer length, and so we can now analyze how well workers who were provided with medium and long solutions performed. To do so, we selected the subset of workers in the filtered validation condition whose first task was restricted to be between 500 and 800 characters (since the first task was never longer than 800 characters by design), and whose second task was at least 1000 characters (n=34 workers). In Table 7 we refer to this subset of workers as the **filtered medium-long** group.

We find that workers in the filtered medium-long group have a much higher average per task accuracy (0.69) than the example-II condition (0.59), validation-II condition (0.57), and filtered validation condition (0.59). The difference is significant ($p < 0.05$) between the filtered medium-long group and validation-II condition after doing a Bonferroni correction for multiple tests. The effect size of per task accuracy for the filtered medium-long workers as compared to the example-II condition was 0.19 (Cohen's h) and the effect size for per worker accuracy between the two conditions was 0.55 (Glass' Δ). The average time per test task and average strategy length for these workers is also larger than in all three of the actual conditions.

Discussion

The results from Experiment II show that the filtered validation condition outperformed the validation-II condition in per task and per worker accuracies (although not significantly) and workers in the filtered medium-long group had a significantly higher average per task accuracy than workers in the validation-II condition, confirming our third hypothesis.

[8] That is, the LASSO assigned a coefficient of 0 to all other predictors.

Number of Workers (Percent of Workers that Start)				
	Start the HIT	Finish ≥ 1 training task	Finish ≥ 1 test task	Finish all tasks
Example-II	310	239 (0.77)	150 (0.48)	102 (0.33)
Validation-II	330	189 (0.57)	140 (0.42)	95 (0.29)
Filtered Validation	314	195 (0.62)	142 (0.45)	88 (0.28)

Table 6. Number of workers who start each condition in Experiment II and the retention rate at various points.

	Per Task			Per Worker		
	Accuracy	Time (min)	Strategy Length (char)	Accuracy	Total Time (min)	Training Cost
Example-II	0.59	8.66 ± 7.25	550 ± 379	0.59 ± 0.26	42.6 ± 20.0	$0.20
Validation-II	0.57	9.02 ± 6.81	561 ± 362	0.58 ± 0.23	53.5 ± 22.1	$1.00
Filtered Validation	0.59	9.58 ± 7.87	618 ± 415	0.60 ± 0.25	52.4 ± 21.5	$1.00
Filtered Medium-Long	0.69	10.96 ± 10.50	692 ± 424	0.74 ± 0.17	55.4 ± 21.6	$1.00

Table 7. Comparison across conditions in Experiment II on metrics of interest. Mean ± standard deviation is shown. Per task accuracy is a Bernoulli random variable, and hence the standard deviation is nearly 0.5 for every condition. The per worker columns only include workers who do all five test tasks.

Moreover, the results show that the filtered validation condition has equal accuracy, and nearly as high retention, as the example-II condition, confirming our final hypothesis. The improved performance of the filtered medium-long group suggest that refining the solutions to be validated can increase the effectiveness of the validation strategy to even outperform the strategy of training with expert examples. (Note that the effect sizes of the filtered medium-long group over the example-II condition were comparable to those of the example condition over the control in the previous experiment!) In future work, the effectiveness of the validation method may be further improved by learning about more sophisticated filtering strategies than filtering simply by length.

FUTURE DIRECTIONS AND CONCLUSION

In this paper we evaluated the effectiveness of different training strategies for crowdsourcing complex problem solving tasks. In our first experiment, we found that expert examples are the most effective form of training among strategies considered in terms of various metrics including worker accuracy. We then showed that having workers validate crowdsourced solutions that are beyond a threshold length is potentially even more effective than reading through expert examples. Our studies primarily focused on the relative efficacy of various forms of training, but we see value in gaining deeper insights on when and why different training strategies work. We briefly looked at this for the gold standard condition, where we identified that this condition helps correct a particular form of incorrect answers that were commonly given by workers. We see multiple directions for future work to further address this question. For example, it is not clear whether the training benefit of the validation condition comes from reviewing peers' solutions or validating them. We hypothesize that the validation process presenting workers a rubric of what the requester is looking for is useful. This was also seen in the work of Dow et al. where workers who self-assessed their work or had an expert assess their work had similar performance gains—both groups saw an almost identical rubric [7]. In that case, would asking workers to "validate" expert examples be an even more effective form of training? It would also be interesting to see to what extent documenting strategies helps workers achiever higher accuracy; do our results hold if we no longer have workers document their strategies after training?

Finally, we think the most practically important future direction is to run similar experiments across a series of complex problem-solving domains to find how generalizable our results are. In particular, it would be interesting to find if filtering by solution length is effective in all domains, and if not, if we can find a general machine learning protocol for finding the features of high-quality validation tasks in any domain. If this is possible, it enables creating crowdsourcing platforms that automatically find how to optimally train unskilled workers. We believe such a pipeline could also be of benefit to the broader educational literature, allowing us to teach problem-solving techniques without having to know them ourselves.

ACKNOWLEDGMENTS

This research was initiated during an internship by Shayan Doroudi at Microsoft Research. The research was supported in part by Carnegie Mellon University's Program in Interdisciplinary Education Research (PIER) funded by grant number R305B150008 from the US Department of Education. We thank Dan Russell for providing several search queries developed for agoogleaday.com.

REFERENCES

1. Scott Bateman, Jaime Teevan, and Ryen W White. 2012. The search dashboard: how reflection and comparison impact search behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1785–1794.

2. Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 313–322.

3. Paul Boekhout, Tamara Gog, Margje WJ Wiel, Dorien Gerards-Last, and Jacques Geraets. 2010. Example-based learning: Effects of model expertise in relation to student expertise. *British Journal of Educational Psychology* 80, 4 (2010), 557–566.

4. Justin Cheng, Jaime Teevan, Shamsi T Iqbal, and Michael S Bernstein. 2015. Break it down: A comparison of macro-and microtasks. In *Proceedings of CHI*.

5. Laura B Cohen. 2001. 10 tips for teaching how to search the Web. *American Libraries* (2001), 44–46.

6. Mira Dontcheva, Robert R Morris, Joel R Brandt, and Elizabeth M Gerber. 2014. Combining crowdsourcing and learning to improve engagement and performance. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 3379–3388.

7. Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM, 1013–1022.

8. Juan M Fernández-Luna, Juan F Huete, Andrew MacFarlane, and Efthimis N Efthimiadis. 2009. Teaching and learning in information retrieval. *Information Retrieval* 12, 2 (2009), 201–226.

9. Morgan Harvey, Claudia Hauff, and David Elsweiler. 2015. Learning by Example: training users with high-quality query suggestions. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 133–142.

10. Pamela J Hinds, Michael Patterson, and Jeffrey Pfeffer. 2001. Bothered by abstraction: the effect of expertise on knowledge transfer and subsequent novice performance. *Journal of applied psychology* 86, 6 (2001), 1232.

11. Slava Kalyuga, Paul Chandler, Juhani Tuovinen, and John Sweller. 2001. When problem solving is superior to studying worked examples. *Journal of educational psychology* 93, 3 (2001), 579.

12. Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 1301–1318.

13. Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. 2011. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 43–52.

14. Avraham N Kluger and Angelo DeNisi. 1996. The effects of feedback interventions on performance: a historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological bulletin* 119, 2 (1996), 254.

15. Andreas Lachner and Matthias Nückles. 2015. Bothered by abstractness or engaged by cohesion? Experts explanations enhance novices deep-learning. *Journal of Experimental Psychology: Applied* 21, 1 (2015), 101.

16. Ard W Lazonder. 2003. Principles for designing web searching instruction. *Education and Information Technologies* 8, 2 (2003), 179–193.

17. Wendy Lucas and Heikki Topi. 2004. Training for Web search: Will it get you in shape? *Journal of the American Society for Information Science and Technology* 55, 13 (2004), 1183–1198.

18. Tanushree Mitra, CJ Hutto, and Eric Gilbert. 2015. Comparing Person-and Process-centric Strategies for Obtaining Quality Data on Amazon Mechanical Turk. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1345–1354.

19. Neema Moraveji, Daniel Russell, Jacob Bien, and David Mease. 2011. Measuring improvement in user search performance resulting from optimal search tips. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 355–364.

20. Fleurie Nievelstein, Tamara Van Gog, Gijs Van Dijck, and Henny PA Boshuizen. 2013. The worked example and expertise reversal effect in less structured tasks: Learning to reason about legal cases. *Contemporary Educational Psychology* 38, 2 (2013), 118–125.

21. David Oleson, Alexander Sorokin, Greg P Laughlin, Vaughn Hester, John Le, and Lukas Biewald. 2011. Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing. *Human computation* 11, 11 (2011).

22. Ron JCM Salden, Kenneth R Koedinger, Alexander Renkl, Vincent Aleven, and Bruce M McLaren. 2010. Accounting for beneficial effects of worked examples in tutored problem solving. *Educational Psychology Review* 22, 4 (2010), 379–392.

23. Adish Singla, Ilija Bogunovic, Gabor Bartok, Amin Karbasi, and Andreas Krause. 2014. Near-Optimally Teaching the Crowd to Classify. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 154–162.

24. John Sweller and Graham A Cooper. 1985. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction* 2, 1 (1985), 59–89.

25. Kurt VanLehn. 1996. Cognitive skill acquisition. *Annual review of psychology* 47, 1 (1996), 513–539.

26. Henry M Walker and Kevin Engel. 2006. Research exercises: immersion experiences to promote information literacy. *Journal of Computing Sciences in Colleges* 21, 4 (2006), 61–68.

27. Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. 2012. Strategies for crowdsourcing social data analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 227–236.

28. Haoqi Zhang, Edith Law, Rob Miller, Krzysztof Gajos, David Parkes, and Eric Horvitz. 2012. Human computation tasks with global constraints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 217–226.

29. Haiyi Zhu, Steven P Dow, Robert E Kraut, and Aniket Kittur. 2014. Reviewing versus doing: Learning and performance in crowd assessment. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 1445–1455.