Communitysourcing: Engaging Local Crowds to Perform Expert Work Via Physical Kiosks

Kurtis Heimerl[1], **Brian Gawalt**[1], **Kuang Chen**[1], **Tapan S. Parikh**[2], **Björn Hartmann**[1]
University of California, Berkeley – Computer Science Division[1], School of Information[2]
{kheimerl,gawalt,kuangc,bjoern}@cs.berkeley.edu, parikh@ischool.berkeley.edu

ABSTRACT

Online labor markets, such as Amazon's Mechanical Turk, have been used to crowdsource simple, short tasks like image labeling and transcription. However, expert knowledge is often lacking in such markets, making it impossible to complete certain classes of tasks. In this work we introduce an alternative mechanism for crowdsourcing tasks that require specialized knowledge or skill: *communitysourcing* — the use of physical kiosks to elicit work from specific populations. We investigate the potential of communitysourcing by designing, implementing and evaluating *Umati: the communitysourcing vending machine*. Umati allows users to earn credits by performing tasks using a touchscreen attached to the machine. Physical rewards (in this case, snacks) are dispensed through traditional vending mechanics. We evaluated whether communitysourcing can accomplish expert work by using Umati to grade Computer Science exams. We placed Umati in a university Computer Science building, targeting students with grading tasks for snacks. Over one week, 328 unique users (302 of whom were students) completed 7771 tasks (7240 by students). 80% of users had never participated in a crowdsourcing market before. We found that Umati was able to grade exams with 2% higher accuracy (at the same price) or at 33% lower cost (at equivalent accuracy) than traditional single-expert grading. Mechanical Turk workers had no success grading the same exams. These results indicate that communitysourcing can successfully elicit high-quality expert work from specific communities.

Author Keywords

Crowdsourcing; Kiosks; Ubiquitous Computing; CSCW

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces - Graphical User Interfaces

General Terms

Design, Human Factors, Experimentation

Figure 1. With Umati, the communitysourcing vending machine, users complete tasks on a touchscreen and receive non-monetary rewards.

INTRODUCTION

Crowdsourcing, the division and assignment of tasks to large, distributed groups of online users, has the potential to create new jobs, improve the efficiency of labor markets, and enable a wide variety of new applications. Researchers have demonstrated compelling new systems enabled by crowdsourcing, including applications that assist the blind with visual tasks [8] and that help writers to copy-edit prose [7]. Many crowdsourcing efforts leverage *microtask* markets, which provide platforms for posting and finding short tasks – frequently seconds to minutes long. One of the best-known markets, Amazon Mechanical Turk (MTurk), attracts thousands of employers [15] and has had hundreds of thousands of worker accounts [30].

One limitation of microtask markets is the difficulty of accessing groups with domain-specific skills or knowledge. The short work durations and small rewards attract specific user demographics [30], limiting the variety of knowledge and expertise available. In contrast, vertical online communities (such as graphic design site 99Designs [1] and Q&A site Stack Overflow [25]) successfully gather experts, but they must either focus on building and maintaining a community of volunteers [25], or on enticing experts with high rewards for complex tasks. These hurdles limit the potential for crowdsourcing of expert work.

To enable crowdsourcing of expert work for short duration tasks, we introduce the concept of *communitysourcing*. Communitysourcing deploys tasks on physical kiosks in specific locations that attract the right "crowds" (while repelling the wrong ones), and in contexts where people have

Figure 2. In our study, Umati was deployed in a public hallway. Students and other building occupants graded CS exam questions on Umati.

idle time ("cognitive surplus" [31]). Communitysourcing leverages the specific knowledge and skills of the targeted community; in return, it provides context- and community-specific rewards that users value more than money.

To explore the potential of communitysourcing, we asked the following research questions:

- Can communitysourcing successfully enlist new user groups in crowd work?
- Can communitysourcing outperform existing crowdsourcing methods for expert, domain-specific tasks?
- How does communitysourcing compare to traditional forms of labor, in terms of quality and cost?

We explored these questions by designing, implementing and evaluating a specific communitysourcing instance: *Umati: the communitysourcing vending machine* (see Figure 1). We built Umati by modifying a commercial vending machine. On Umati, instead of inserting money, users select and perform tasks on a high-resolution touch screen. Earned credits can be used to choose and buy items — an internal computer controls the motors that operate each of the vending arms.

We tested Umati with a Computer Science exam grading task. Grading exams is a time consuming, high-volume job that requires significant domain expertise for consistent, correct scoring of open-ended questions. Grading can also be partitioned into many small tasks. We placed Umati in the primary hallway in our Science building and filled it with snacks (see Figure 2). Exams covered introductory Computer Science topics in programming languages, algorithms, and data structures. Umati successfully targeted a specific population (81% of users were trained in Computer Science or Electrical Engineering) and achieved significant throughput: 328 participants graded 7771 exam answers in one week, for approximately $200 of snacks. Umati attracted new workers: 80% of participants had never participated in crowdsourcing before. We compared communitysourced grades against grades from a set of ten experts. Umati was able to exceed the accuracy of traditional single-expert grading (80.3% to 78.3% at the same price) or, alternatively, lower cost by 33% (at the same accuracy). For comparison, we also recruited graders on Mechanical Turk; these workers could not grade

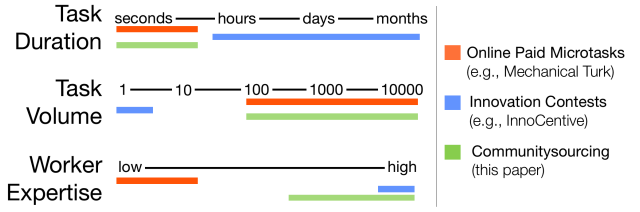

Figure 3. Communitysourcing occupies a unique position in the design space of crowdsourced work: it enables short, high-volume tasks that require high levels of expertise.

accurately ($< 25\%$ agreement with our experts). Our results suggest that communitysourcing can exceed traditional expert performance in domains where online crowdsourcing fails.

In this paper, we begin with a discussion of related work and describe the design space of communitysourcing. We motivate the choice of an exam grading task, then describe the implementation of Umati. We follow with a detailed evaluation of the system and conclude with a summary of our results and pointers to future work.

RELATED WORK

Communitysourcing touches on prior work in crowdsourcing, location-based computing, and kiosk systems. Our experimental task also relates to prior efforts in peer grading. We discuss each area in turn.

Crowdsourcing

Crowdsourcing and human computation are active areas of research inquiry; recent surveys describe a range of approaches [21, 28]. Projects have investigated incentives and game mechanics to motivate workers [35], new applications enabled by crowdsourcing [7, 8, 36], tools to develop crowdsourcing applications [22], and algorithms to coordinate crowd work [5, 6, 20]. Researchers have also contributed methodologies for conducting experiments using online labor [13, 19]. Much recent work focuses on increasing the complexity of work products created by a pool of unskilled workers through redundant microtasks. While communitysourcing also targets tasks of short duration, it recruits domain experts through appropriate choice of location and incentive (see Figure 3).

The dominant strategy to elicit expert responses or expert work online has been to build narrow, domain-specific communities. Topical communities of unpaid volunteers can provide high-quality answers to technical questions. However, creating successful online communities is an uncertain, labor-intensive proposition that requires leadership and constant monitoring [25]. Some online platforms succeed in eliciting paid expert work through competitions, e.g., InnoCentive for pharmaceutical research [2]; 99Designs for graphic design [1]; or TopCoder for programming [4]. Competitions are not appropriate for high-volume work because they strive to find the single best solution among the submissions. Communitysourcing can provide higher throughput by engaging many experts in short tasks.

Location-based Computing and Kiosks

Researchers have investigated the role of physical location in crowdsourcing. In participatory sensing [12], people collect and report data about their environment. For example, Creek Watch [18] uses GPS coordinates to crowdsource local watershed measurements. Multiple mobile start-up companies are delivering questions and tasks to members in a given location [11, 23]. In these systems, users first have to sign up for the service and agree to have their location tracked. Communitysourcing attracts people who already frequent a particular locale through a kiosk interface.

Computer-based kiosks have long been used for self-service ticketing, banking, and location-specific information, e.g., in museums [17]. Kiosks have also been deployed in rural areas and developing countries for Internet access, for unsupervised education [27, 32], and to aid communities in reconciliation after civil unrest [33]. Research has focused on security of such public systems [9] and on facilitating information transfer between kiosks and other computing devices [14, 16]. To our knowledge, Umati is the first kiosk that elicits work and rewards workers with physical items.

Education and Grading

Grading open-ended exams is an intrinsically subjective process. One common mechanism for consistently and efficiently grading open-ended questions is peer assessment. Students grading other students can lead to positive impacts on learning and attitude [24, 34]. Peer assessment is usually restricted to students within a particular course, and is prescribed by course staff. In Computer Science, prior work has investigated web-based peer review of assignments [10]. Averages of redundant peer scores are strongly correlated with teachers' scores [29]. Umati generalizes peer assessment beyond course boundaries: it enlists volunteers and provides physical rewards to incentivize participation.

DESIGN CONSIDERATIONS FOR COMMUNITYSOURCING

Successful communitysourcing depends on careful selection of tasks, locations, and incentives. This section discusses important design considerations for each category.

Task Selection

Communitysourcing is best suited for short-duration, high-volume tasks that require specialized knowledge or skills which are specific to a community, but widely available within that community. Simple tasks without expertise requirements (e.g., address verification or text transcription) are a better fit for generic crowdsourcing platforms like Mechanical Turk. Low-volume, long-duration expert jobs (like graphic design or programming) are better served by vertical online markets.

Some representative tasks well-suited for a communitysourcing approach include grading exams in academia, bug and tech support triage in technology companies, fact checking for journalism, and community-specific market research in bars or specialty shops. Short task durations enable users to work on a kiosk without ergonomic issues, and enable many user to participate. Public kiosks can potentially attract a large number of users, supporting, as well as requiring, a high task volume.

Location Selection

A communitysourcing system must target those locations where the required skills are prevalent. The locations must also repel those *without* the required knowledge. For instance, placing a kiosk in a bar could potentially target young men with market research tasks – but the kiosk would also be accessible by young women. A better location to exclusively target men would be their restroom at the bar.

The targeted community must also have some "cognitive surplus" [31] available for doing work. Practically, this means placing the kiosk in an area where people have spare time, e.g., in lounges, government office waiting rooms, or airports.

Reward Selection

Rewards must be interesting and valuable to the targeted community in the given location. For instance, airport travelers may value Wifi Internet access. In contrast, Wifi access has lower utility on a college campus, where free Internet is readily available. While cash is an obvious reward (consider an ATM-like kiosk), it has been shown that individuals can value items more than money in specific contexts and locations [26].

COMMUNITYSOURCING IN ACTION

For our deployment, we chose exam grading as an experimental task, a university department as the location, and snacks as rewards. This section briefly justifies these design choices in light of the earlier design guidelines for communitysourcing.

Task: Grading exams is a painful, high-volume task for teaching staff. An individual student answer can be graded quickly, but large courses with hundreds of students generate thousands of answers. While people outside an academic discipline are unlikely to possess the requisite knowledge to grade open-ended problems, redundant peer grades are strongly correlated with expert scores [29], suggesting that students can grade each others' assignments and exams.

Location: College campuses commonly assign buildings to disciplines (departments). These buildings rarely attract visitors outside the discipline. Our Computer Science (CS) building supports mainly CS classes, along with a few unrelated events. We placed our kiosk in front of the major lecture hall in the CS building, an area where students often wait for class to begin, maximizing the cognitive surplus.

Reward: Food is the most common reward given to entice students to participate in campus events. Companies commonly provide pizza to bring students to recruitment events; teachers bring candy to class sections to entice participation. We decided to use snacks as our reward. This choice also impacted the physical design of our kiosk. The most common system for distributing snacks is the ubiquitous vending machine; it is familiar and clearly communicates the nature of the rewards being distributed.

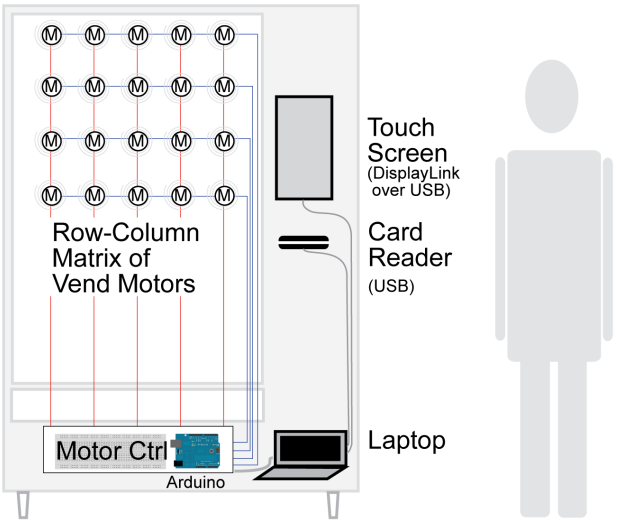

Figure 4. Umati uses an internal laptop to control a touchscreen, USB card reader, and the vending motors. Users complete tasks on the touchscreen for rewards from the machine.

These design decisions guided our implementation of *Umati: the communitysourcing vending machine*, as described in the next section.

UMATI: SYSTEM IMPLEMENTATION

The Umati system comprises custom vending hardware; a generic software platform for work tasks; and specific task user interfaces. We describe each in turn.

Hardware: Hacking a Vending Machine

Umati is a modified commercial vending machine (specifically, a 1986 Snackshop 4600, Figure 1). We removed all legacy vending equipment (the bill exchanger, coin reader, keypad interface, analog motor-control circuit boards), leaving just the vending shelves and dispensing mechanism. We installed a touchscreen and keycard reader on the front of the machine, which provide the primary user interface (see Figure 4). Users must first swipe their magnetic strip ID card (given to all students, faculty, and staff on campus); card IDs enable Umati to distinguish users, though our prototype does not link card IDs to users' names. Users complete tasks on the touch screen interface which is connected to an internally mounted laptop.

Rewards are dispensed through programmatic control of the vending mechanisms. The shelves use metal spirals driven by DC motors. The motor leads form a row-column matrix: when a particular row lead is powered and a column lead grounded, the motor at the intersection of row and column spins, dispensing an item. We attached the row-column end points to relays controlled by an Arduino microcontroller, and programmed the microcontroller to respond to vend requests from the laptop over a serial link. Motors spin uniformly enough to vend items with open-loop control, powering motors for a fixed amount of time.

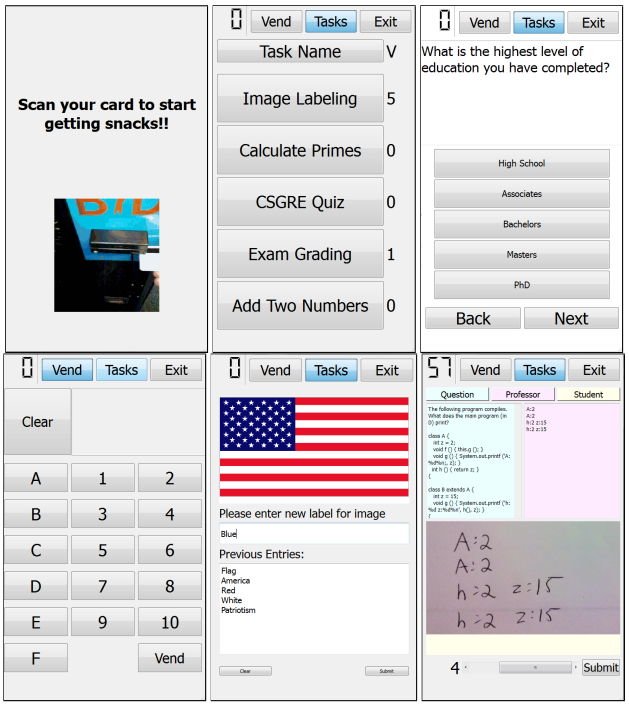

Figure 5. Umati's UI, clockwise from top left: introductory attract screen, task selection, survey task, grading task, image labeling task, and vending interface.

Software: Maximizing a Small Touch Screen

We developed the Umati software using Python and the PyQt4 UI toolkit. Umati's architecture was designed as a generic task platform: each task and its user interface are written as plug-ins to a work management system. As a result, Umati supports many task types besides exam grading.

When the machine is idle, it displays an *attract screen*: a looping video demonstrating how to log in by swiping an ID card (see Figure 5). After swiping, users are presented with the *Task Selection* screen, allowing them to choose their task. If there is just one available task, the selection screen is bypassed and users are immediately presented with that task. The user's earned credits are shown in the top left, and stored across work sessions. When a user wishes to cash in their credits, they switch to the *Vending* interface which provides a software keypad to select the tray of the desired item.

Tasks

We have implemented a variety of tasks for Umati: math tests, CS GRE questions, surveys, exam grading, and image labeling. Some of these task interfaces are shown in Figure 5. For our evaluation, we tested just two: surveys and grading.

Surveys: The survey task constructs an interface from an external XML file that defines questions and multiple choice answer sets. For our deployment, Umati had a basic demographic survey which collected age, department, educational status, and prior exposure to crowdsourcing. Every user was required to complete this survey before starting another task on Umati.

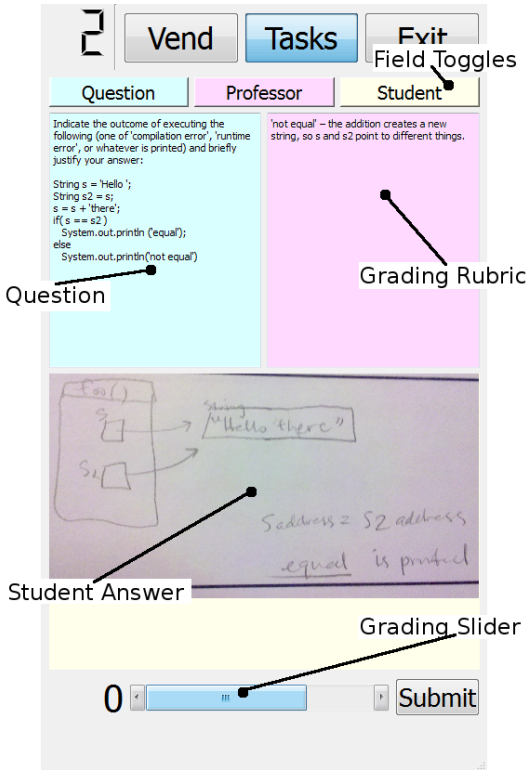

Figure 6. The Umati grading interface shows a question, an answer key, and an image of a student response. Workers assign points with a slider.

Grading: The grading interface presents three panels of information simultaneously to the participant: the question, the instructor's answer key, and a student answer (see Figure 6). Each panel can be opened and closed using toggle buttons on the top row, to maximize the limited screen space. Each panel also allows panning by dragging and zooming by double-tapping. Grades are gathered on a numeric scale: the user enters the grade using a slider widget. Upon submission, Umati presents another student answer *to the same question*, if one is available. This allows a user to memorize the question and answer key, accelerating their grading of student answers. It also supports grading privacy, as graders cannot easily grade all of a specific student's answers. Users cannot grade the same exam answer twice.

Spam Detection: Incorrect or low-quality responses ("spam") are common for crowdsourcing; workers often complete tasks quickly and with little thought in order to maximize their earnings. To account for this tendency, Umati implements a simple spam detection system. Each task definition includes "gold standard" tasks, comprised of a task with a known correct response. If a user fails two of these tasks, that user is logged out and their ID is blacklisted from future use. Inserting such checks is a well-established technique for improving the quality of crowdsourced work [19].

EVALUATION

Our evaluation sought to answer three primary research questions about communitysourcing: Can it enlist new user groups in crowd work? Can it outperform existing crowdsourcing methods? And can it match more traditional approaches, in terms of cost and accuracy?

Study Design

We created a set of Computer Science exams with answers. Three groups of participants graded the exams: users of our vending machine (Umatiers), Mechanical Turk workers (Turkers), and a group of former or potential Computer Science teaching assistants (Experts).

Exam Corpus

We created a custom exam from prior mid-term questions of our Computer Science department's second-semester undergraduate course, *CS2*. The exam was comprised of 13 questions covering basic complexity theory, object-oriented programming, search algorithms, and bit operations. Eight undergraduate students were recruited as test-takers. These students had recently completed CS2 or were enrolled at the time of the study. Students completed the exam on paper. For our *gold standard* tasks, we added an exam completed using the instructor's grading rubric, giving us question/answer pairs that are always correct. These answers were used for our spam detection algorithm. 105 sample exam answers were generated in total. Test-takers received either a 5 USD gift card, or could make an equivalent charitable donation.

Common Grading Instructions

Graders in all conditions were asked to assign 0 to 4 points for each question. The Instructor's answer key was used as the grading rubric. The rubric provided no instructions for assigning partial credit. While all questions required computer science knowledge to produce a correct answer, they varied in the amount of knowledge required to assess that answer. Some were *simple*, requiring only basic pattern matching for identifying correct answers. Others were *open-ended*, requiring a deeper understanding of the underlying material. Assigning partial credit for incomplete answers always required domain-specific knowledge.

For the exam we tested, five questions were simple, while eight were open-ended. An example open-ended question and grading rubric follows:

> *Q: Briefly explain the difference between a (Java) instance variable and a (Java) class variable.*
>
> *A: Each object has its own distinct copy of an instance variable. But all the objects in a class share just one 'copy' of a class variable.*

Paper exams had a defined answer box for each question. We used Shreddr [3] to extract scanned answers from the exams. These images of answers were shown in the grading interface (Figure 6). Three sets of participants then graded the exams: 1) Umati users, 2) Mechanical Turk users, and 3) experts recruited from a pool of qualified students. These conditions are described in more detail below.

Grading on Umati

We deployed Umati outside the main Computer Science lecture hall for one week. This lecture hall is primarily used for undergraduate courses, although graduate students, staff, faculty and visitors also frequent this centrally located ground-floor area. Users received no other introduction to the system or tasks besides the looping video instructing them to swipe their card.

The machine was loaded with $200 worth of candy. Users earned one credit per graded answer, and five for completing the survey. Most items were priced at 20 credits (with some priced at 10 and 30). In a small pilot study, we found that users completed roughly 20 tasks in five minutes. We estimated that five minutes was the maximum amount of time users would be willing to spend in front of a vending machine at any one time. Candy was purchased from a local bulk wholesale store at an average of 45 cents per item. At this price, Umatiers were paid 2.25 cents per answer, in candy.

Grading on Mechanical Turk

Workers on Amazon's Mechanical Turk (Turkers) graded the same exam questions and answers, on a web interface that was similar to the Umati touch screen interface; workers saw the question, the answer key, and the student's answer. Each Human Intelligence Task (HIT) consisted of grading all 9 answers to the same question (8 students and the experimenter's gold standard), paying 23 cents (2.6 cents / answer). We posted 130 HITs on Turk, obtaining 10 sets of grades for each student answer.

Turkers answered the same gold standard questions as Umati users, and we applied a similar spam detection algorithm. We present two sets of results: 1) all Turkers (regardless of gold standard performance); and 2) the subset of Turkers that passed the spam filter (failing at most one gold standard question). As with Umati, any answers provided before a second failed question are used in our analysis.

After a first set of workers completed our grading HITs, we also attempted to recruit more knowledgeable Turkers through a qualification test. Workers had to pass the test before they could grade exam answers. The test comprised five multiple choice questions on computational complexity and Java. For example:

What is the Big-O run-time of the following algorithm?

```
function(arg1, arg2):
   return arg1+arg2
```

A: $O(1)$ *B:* $O(logn)$ *C:* $O(n)$ *D:* $O(nlogn)$ *E:* $O(n^2)$

For these qualified Turkers we offered 25 cents per set of 9 grades (2.8 cents /answer).

Grading By Experts

Ten former or potential CS teaching assistants graded the exam corpus on paper. These graders included: one graduate instructor for *CS 2*, one high-scoring undergraduate (received an A in *CS 2*), and eight CS PhD students. All had prior teaching experience. Each participant was paid 25 USD in Amazon credit to grade all 105 answers.

At our university, only graduate students are allowed to grade exams. Graduate students earn over 38 dollars per hour, when including basic tuition and fee remission. On average, experts graded 115 answers per hour, for an effective rate of 34 cents per answer. All of the expert graders passed the gold-standard spam detection algorithm, as should be expected.

Measures

Grading is inherently subjective — even experts can disagree. To enable meaningful comparisons between our different conditions, we define dependent measures that compare *grade distributions* and *agreement with median expert scores*.

Comparing Grade Distributions

For both crowdsourced grading methods, Umati and Mechanical Turk, we compute a Chi Squared test statistic across the 105 exam answers to determine if the distribution of crowdsourced grades in either condition differs significantly from the expert distribution.

Grade Agreement

To measure the grading accuracy of our approach, we use the median of all expert graders (*median expert grade*) as the "correct" grade for a student's answer. We then compute *grade agreement* as the percentage of exam answers for which the median Umati and Turk grades match the median expert grade.

The status quo is to use a single expert grader, not the median of multiple graders. To investigate the relationship between grading accuracy and the cost of recruiting additional graders, we randomly sample (with replacement) subsets from each distribution of graders (expert, Umati, and Turk). For instance, we sample a single expert grader from the expert distribution, and compare that to the median expert grade for all exam answers. Similarly, we also sample subpopulations (for instance, seven Turkers), take the median of their grade distribution, and compare that to the ("correct") median expert grade for all exam answers. This is repeated 1000 times to compute the *subgroup grade agreement* (e.g., between seven Turkers and all ten experts).

Results

We report descriptive usage results, comparisons of the experimental conditions, and qualitative observations.

Umati Users

328 participants graded 7771 exam answers on Umati in one week. Figure 7 shows self-reported demographic data from the survey. The users were primarily undergraduate computer scientists. 80% of Umati users had never previously participated in any online crowdsourcing activity (showing that Umati engaged new users in crowdsourcing). Users who successfully completed at least one task had a median usage time of 4 minutes, completing an average of 16 tasks. The most active user (or group, see below) worked continuously for more than an hour, completing 85 tasks in a row. Several users completed all 105 of the grading tasks. 61 users (19%) were blacklisted by the system for failing gold standard tasks.

Filters	Participants	%	Tasks Completed	Task %
Total	328	100%	7771	100%
Undergraduates	263	80%	6297	81%
Graduates	39	12%	943	12%
Other (Professors, Visitors, Staff)	26	8%	531	7%
Computer Science	234	71%	5977	77%
Electrical Engineering	34	10%	485	6%
Other/Not Affiliated	60	18%	1309	17%
Had Not Crowdsourced	262	80%	5933	76%
Had Crowdsourced	66	20%	1838	24%
Knows about Crowdsourcing	216	66%	4469	58%
Does Not Know About Crowdsourcing	112	34%	3302	42%

Figure 7. Descriptive statistics for one week of Umati use.

The high failure rate may be a consequence of the machine's novelty: users may have explored the interface without understanding the consequences of incorrect answers.

81% (268) of Umati users majored in Computer Science or Electrical Engineering; both groups are required to take *CS 2*. Our system was able to effectively engage our target population and complete a large number of tasks. This demonstration of Umati's ability to target crowdsourcing work to hundreds of unique, expert users over the span of just one week is a primary result of our work. We also note that these numbers were achieved despite limited resources: twice, our vending machine was emptied of snacks in less than two days; we refilled once. Effectively, the machine ran for four of the seven days.

Mechanical Turk Users

We gathered 1050 grades in three days using Mechanical Turk. 46 unique Turkers graded exams, with 16 (35%) failing the spam detection (incorrectly grading two or more *gold standard* questions). While we offered higher wages to workers who passed the CS qualification, no workers successfully qualified during our experiment. We conclude that a qualification exam is not an efficient mechanism for attracting workers with computer science knowledge on Mechanical Turk.

Comparing Grade Distributions

We computed the distribution of grades for each exam answer from experts, Turkers and Umatiers. These distributions were summed across all responses, and across all answer items, to produce the statistic used to test the potential differences between response distributions.

Chi Squared tests find a significant difference between the expert and either Turker score distribution ($P_{TURK} < .001$ and $P_{TURK_{NoSpam}} < .001$). We see no such significant difference between the experts and Umati users ($0.263 < P_{UMATI} < .997$) [1]. This suggests that the Umatiers, on aggregate, grade similarly to experts; Turkers do not.

[1]The 105 survey items allowed for 5 score options, zero to four. Each individual answer's distribution then has 4 degrees of freedom (knowing the probability mass in bins 0 through 3 fixes the amount of mass in bin 4). From this, we know the number of degrees of freedom for the overall Chi Squared test is at least $4 \times 105 = 420$ and at most $5 \times 105 - 1 = 524$, depending on the amount of dependence between the student answers. This provides us with bounds on the p-values of each experiment.

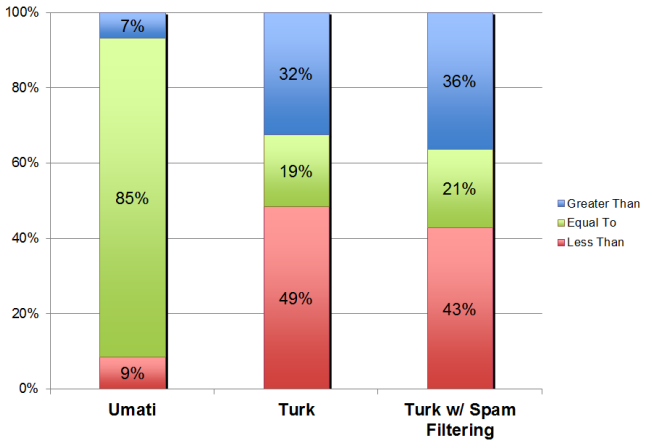

Figure 8. Agreement between conditions and the median of the 10 expert graders. Umati users' response distributions were much closer to the experts than distributions of Mechanical Turk users.

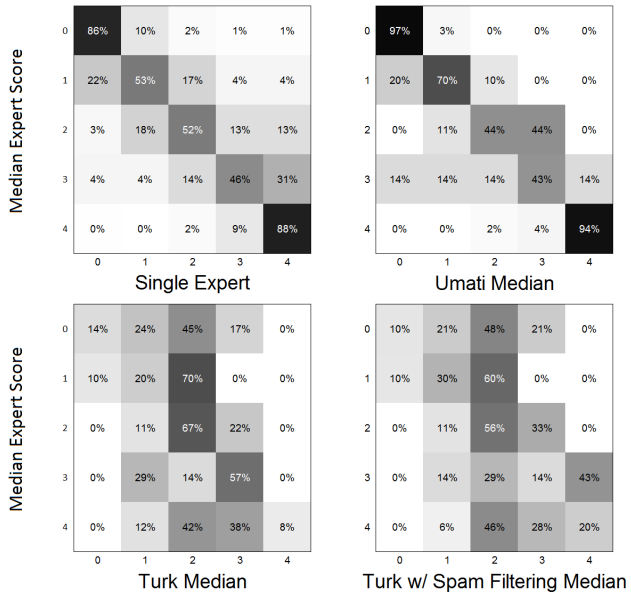

Figure 9. Comparing grade agreement between the the experts and our test conditions. Umatiers and single experts often agreed with the pool of experts. Turkers were consistently random.

Comparing Grades

Umati users agreed with the experts' median grade 85% of the time (Figure 8). In comparison, Turkers only agreed with experts 19% of the time; this figure increases only slightly to 21% with spam filtering. [2]

Figure 9 shows how different graders diverged from the grades of *all* expert graders. Boxes along the diagonal show agreement with experts. Individual experts and Umatiers agree strongly with all experts on assigning full credit or no credit. Single experts appear to have less variance than

[2]Random guesses have an expected agreement of 20% on a five-point scale.

Umatiers when assigning partial credit. We reserve a careful study of systematic biases to future work. MTurk workers' responses have no discernible pattern, leading to a generally centered (2/4) median score.

Price-Accuracy Trade-off

We explore the effects of assigning varying numbers of graders on accuracy by sampling subgroups from the distribution of experts, Turkers, and Umatiers and calculating median agreements for these groups to all experts.

Figure 10 plots agreement as greater numbers of respondents are considered. Individual expert graders agreed with the median of all experts 78.3% of the time. Ten Umatiers agreed 79% of the time, but cost 33% less ($10 \times 2.3 = 23$ cents/grade vs 34 cents/grade). Mechanical Turk users never approximate the experts, regardless of their number. In fact, Turkers are *less* likely to agree with the median expert grade as the number of respondents increases.

If we keep cost constant, Umati is more accurate than a single expert grader (see Figure 11). Thirty-four cents can buy one expert answer, with an accuracy of 78.3%; or fifteen Umatiers, who have an accuracy of 80.3%. Because result quality is achieved through multiple graders at lower individual pay, it is easy to increase or decrease the accuracy by adjusting the number of redundant assignments. Additional workers can also be dynamically assigned to more difficult or controversial cases where initial ratings diverge, while the number of assignments can be held low for answers that are clearly correct or incorrect.

Qualitative Observations

Throughout the week-long deployment, we observed users to obtain feedback on usability and user experience. At times, student interest was higher than our single machine could handle: queues formed in front of the machine. We noticed that groups of students often used the system together (see Figure 2). Some groups were formed ad-hoc, by students waiting in line. Other groups approached the machine together. Using Umati seemed to be a social event; groups would argue about the specific merits of different answers. Such additional discussion is an unexpected secondary benefit: academic departments will likely welcome increased discussion of class material by their students. Groups would also share the rewards, for instance splitting a bag of candy amongst the participants.

However, groups could also negatively impact grading: student groups seemed more inclined to attempt to defraud the system. One group attempted to pry open the door. Queues of waiting students potentially limit the throughput of the machine, but are hard to avoid — physical interfaces do not scale easily. A number of users complained that they were unjustly blacklisted. Often this was because they had forgotten to log out. A few users reported being removed because they were exploring and were not aware of the repercussions. Better instructions about safeguarding mechanisms could address these problems.

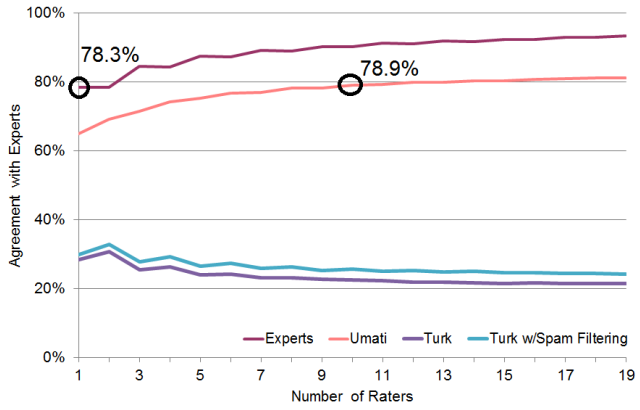

Figure 10. The accuracy of Umati users becomes comparable to a small number of experts as more participants are involved. Here, we show the point where Umati accuracy surpasses that of a single expert.

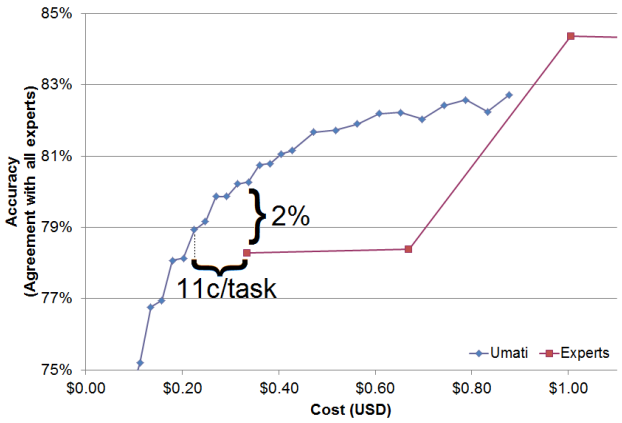

Figure 11. Accuracy (compared with all experts), for experts and Umati. One expert grader costs the university approximately 34 cents/task.

DISCUSSION

In this section we discuss some important limitations of Umati and of our particular study design. We also discuss possible alternative strategies for communitysourcing.

Privacy and Confidentiality for Grading

Publicly displaying student answers raises obvious privacy and confidentiality concerns. For example, handwriting might reveal the identity of a test taker. Students may not want peers grading their exams (though peer grading already exists in different forms at many schools). For the experiment described in this paper, paid participants generated exam answers; students' course grades were not determined by Umati workers. In the future, Umati can be used in situations where students have consented to this form of grading at the outset of a course. Alternatively, Umati can be used for many forms of assessment with lower stakes, e.g., homework grading or practice exams.

With the current design, students may also attempt to game the system by providing high grades to their own (or their friend's) answers. We limited this concern by forcing graders to view all submissions to one question before moving on to the next questions. Attackers thus must grade *every* answer to *every* question in order to comprehensively cheat for just one student. Umati also gathers the student's ID, allowing us to programmatically block them from grading exams for classes they are enrolled in, or have not yet taken. To reduce the risk of cheating and of inadvertent identification by peers, test-takers and test-graders could also be geographically and socially separated – for example, students on the East Coast could grade assignments from students on the West Coast, and vice versa.

Study Limitations

Our current study has several important limitations that should be addressed in future work.

No Comparison to Expert Work Platforms

We compared the effectiveness of grading tasks on Umati to Mechanical Turk. However, other crowdsourcing technologies could potentially be useful for grading. For instance, the computing experts on Stack Overflow [25] should have the expertise required to assess introductory exams. Similarly, we could hire a contractor on oDesk.

We are skeptical that these other engines are well suited for short-duration, expert tasks like grading. An individual expert (as on oDesk) is unlikely to be better or cheaper than the student graders we compared against. Stack Overflow workers provide the *best* answer to a question, disallowing the high-volume of tasks needed for grading. However, we have not yet tested whether these intuitions hold.

Short Deployment

A major limitation of our study is the short one-week duration of the deployment. High initial usage led to task starvation: we did not have enough sample tasks for graders. After one week, each exam response had already gathered an average of 68 grades.

While we know that the initial interest was high, we cannot yet comment on sustained, steady-state usage of Umati. We observed that more tasks were completed in the latter half of our deployment, suggesting that user momentum was still building. We are actively seeking new high-volume tasks for future deployments.

Similarly, we cannot say how much of this initial interest in Umati was due to the specific properties of our targeted crowd. It may be that Computer Science students are more excited about crowdsourcing than other groups. We hope to resolve this by targeting different communities (with different tasks) in the future.

Interface Affordances Limit Work Duration

We chose grading as an example communitysourcing task for Umati in part because it could be achieved with minimal user input. Umati's design might preclude many other types of work. This is primarily due to its limited input and output affordances: a 7-inch vertical touch-screen is inefficient and error-prone for many data-entry heavy tasks. Kiosks must appear *approachable* to attract users and be *robust* to public abuse. Overly complex interfaces can create challenges for achieving these goals.

An alternative would be to divide work and reward interfaces: experts could perform work on their own computers or smart phones, e.g., through a web interface, and then come to a nearby kiosk or vending machine to redeem their rewards. In such a design, the kiosk would primarily enforce locality and community membership. However, this design would reduce the social visibility of the machine: in our deployment, students were frequently attracted by watching other students use Umati. In addition, communitysourcing requires available cognitive surplus. There are few other distractions in a public hallway; spare time may be more difficult to find when competing with a myriad of ways to spend time on the Internet. We reserve further study of these questions for future work.

Applications of Communitysourcing to Other Problems

Our study demonstrated the value of communitysourcing for one particular task domain and reward type. Grading is just one example of a high-volume task that requires an expert population; we could instead provide specialized surveys. Snacks are just one of many possible rewards; students may prefer video game currency to snacks. In the future, we plan to develop alternative embodiments of communitysourcing, and to develop a theory of how the combination of task, location and reward affect work quality and quantity. For example, future deployments could include: a bug triage espresso machine in the lounge of a big technology company; a jukebox with market research tasks at a bar; or a slot machine in the airport with travel review tasks.

CONCLUSION

In this work we introduced the idea of communitysourcing — the use of physical kiosks to crowdsource work from targeted populations. Communitysourcing is achieved by: 1) situating tasks in specific physical spaces that attract the right "crowd" and where people have idle time and 2) providing physical, context- and community-specific rewards that users value more than money. These two key points invite new expert workers to participate in short duration crowd work.

To explore the potential of communitysourcing, we designed, implementing and evaluated a specific communitysourcing interface for a particular expert task. *Umati, the communitysourced vending machine* enabled users to earn credit for purchasing snack items by performing grading tasks on a touchscreen. We investigated whether Umati can recruit new users: over one week, Umati successfully targeted a student community, with 302 students (of 328 total) grading 7240 (of 7771 total) exam answers. 80% of all the users had never participated in crowdsourcing before. We also investigated how task quality compared to both online crowdsourcing and traditional, offline work.

We found that Umati was able to grade exams more accurately (80.3% vs 78.3% at the same cost), or at lower cost (23c/answer vs 34c/answer at equivalent accuracy), than traditional single-expert grading. Workers on Mechanical Turk were unable to successfully grade the exams.

In the future, we would like to test Umati over a longer period of time, to observe patterns of use and investigate the long-term prospects of expert work on kiosks. We would also like to consider alternate communitysourcing scenarios with varying configuration of task, location and reward; for example, one where users could complete tasks online using their own computers, obtaining one-time keys that can be redeemed for credit.

ACKNOWLEDGMENTS

We thank Bryan Trinh for his help constructing Umati and Kehontas Rowe for task design and video production. We thank Kelly Buchanan, Isabelle Stanton, and Alice Chen for putting up with us. We greatly appreciate the help received from Robert Carroll, Barret Rhoden, Anuj Tewari, Kevin Klues, Lisa Fowler, and Wesley Willett in moving the machine. Lastly, we thank Eric Brewer and Laurent El Ghaoui for letting us take on this project.

REFERENCES

1. 99Designs. http://www.99designs.com/. Retrieved 9/2011.
2. Innocentive. http://www.innocentive.com/. Retrieved 9/2011.
3. Shreddr. http://www.shreddr.org/. Retrieved 9/2011.
4. Topcoder. http://www.topcoder.com/. Retrieved 9/2011.
5. Ahmad, S., Battle, A., Malkani, Z., and Kamvar, S. The jabberwocky programming environment for structured social computing. In *Proceedings of UIST'11*, ACM (2011), 53–64.
6. Bernstein, M. S., Brandt, J., Miller, R. C., and Karger, D. R. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of UIST'11*, ACM (2011), 33–42.
7. Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., and Panovich, K. Soylent: a word processor with a crowd inside. In *Proceedings of UIST'10*, ACM (2010), 313–322.
8. Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., Miller, R., Tatarowicz, A., White, B., White, S., and Yeh, T. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of UIST'10*, ACM (2010), 333–342.
9. Garriss, S., Cáceres, R., Berger, S., Sailer, R., van Doorn, L., and Zhang, X. Trustworthy and personalized computing on public kiosks. In *Proceeding of MobiSys '08*, ACM (New York, NY, USA, 2008), 199–210.
10. Gehringer, E. F. Electronic peer review and peer grading in computer-science courses. In *Proceedings of SIGCSE'01*, ACM (2001), 139–143.
11. Gigwalk. http://www.gigwalk.com. Retrieved 9/2011.
12. Goldman, J., Shilton, K., Burke, J., Estrin, D., Hansen, M., Ramanathan, N., Reddy, S., Samanta, V., Srivastava, M., and West, R. Participatory Sensing: A citizen-powered approach to illuminating the patterns that shape our world. *Foresight & Governance Project, White Paper* (2009).
13. Heer, J., and Bostock, M. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of CHI'10*, ACM (New York, NY, USA, 2010), 203–212.
14. Huang, A., Pulli, K., and Rudolph, L. Kimono: kiosk-mobile phone knowledge sharing system. In *Proceedings of MUM'05*, ACM (New York, NY, USA, 2005), 142–149.
15. Ipeirotis, P. G. Analyzing the Amazon Mechanical Turk marketplace. *ACM XRDS 17* (December 2010), 16–21.
16. Izadi, S., Brignull, H., Rodden, T., Rogers, Y., and Underwood, M. Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In *Proceedings of UIST'03*, ACM (New York, NY, USA, 2003), 159–168.
17. Kearsley, G. *Public Access Systems: Bringing Computer Power to the People*. Ablex Publishing Corporation, 1994.
18. Kim, S., Robson, C., Zimmerman, T., Pierce, J., and Haber, E. Creek watch: Pairing usefulness and usability for successful citizen science. In *Proceedings of CHI '11*, ACM (2011), 2125–2134.
19. Kittur, A., Chi, E. H., and Suh, B. Crowdsourcing user studies with mechanical turk. In *Proceeding of CHI'08*, ACM (New York, NY, USA, 2008), 453–456.
20. Kittur, A., Smus, B., Khamkar, S., and Kraut, R. Crowdforge: Crowdsourcing complex work. In *Proceedings of UIST'11*, ACM (2011), 1801–1806.
21. Law, E., and von Ahn, L. *Human Computation: An Integrated Approach to Learning from the Crowd*. Morgan & Claypool, 2011.
22. Little, G., Chilton, L. B., Goldman, M., and Miller, R. C. Turkit: human computation algorithms on mechanical turk. In *Proceedings of UIST '10*, ACM (New York, NY, USA, 2010), 57–66.
23. Localmind. http://www.localmind.com. Retrieved 9/2011.
24. Mafi, M. Grading: Involving students in a time-saving solution to the homework problem. *Engineering Education 79*, 3 (April 1989), 444–446.
25. Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., and Hartmann, B. Design Lessons from the Fastest Q&A Site in the West. In *Proceeding of the CHI'11*, ACM (2011), 2857–2866.
26. Menger, C. *Grundsätze der Volkswirtschaftslehre*. Braumüller, 1871.
27. Mitra, S. Self organising systems for mass computer literacy: Findings from the 'hole in the wall' experiments. *International Journal of Development Issues 4*, 1 (2005), 71–81.
28. Quinn, A., and Bederson, B. Human computation: A survey and taxonomy of a growing field. In *CHI '2011: Proceeding of the twenty-ninth annual SIGCHI conference on Human factors in computing systems* (2011), 1403–1412.
29. Reily, K., Finnerty, P. L., and Terveen, L. Two peers are better than one: aggregating peer reviews for computing assignments is surprisingly accurate. In *Proceedings of GROUP '09*, ACM (2009), 115–124.
30. Ross, J., Irani, L., Silberman, M. S., Zaldivar, A., and Tomlinson, B. Who are the crowdworkers?: shifting demographics in mechanical turk. In *Extended Abstracts of CHI'10*, ACM (2010), 2863–2872.
31. Shirky, C. *Cognitive Surplus: Creativity and Generosity in a Connected Age*. Penguin Press, 2010.
32. Slay, H., Wentworth, P., and Locke, J. Bingbee, an information kiosk for social enablement in marginalized communities. In *Proceedings of SAICSIT '06* (2006), 107–116.
33. Smyth, T. N., Etherton, J., and Best, M. L. Moses: exploring new ground in media and post-conflict reconciliation. In *Proceedings of CHI '10*, ACM (2010), 1059–1068.
34. Topping, K. Peer assessment between students in colleges and universities. *Review of Educational Research 68*, 3 (1998), 249–276.
35. von Ahn, L., and Dabbish, L. Designing games with a purpose. *Communications of the ACM 51* (August 2008), 58–67.
36. Yan, T., Kumar, V., and Ganesan, D. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In *Proceedings of MobiSys '10*, ACM (New York, NY, USA, 2010), 77–90.