Crowdsourcing in HCI Research

Serge Egelman, Ed H. Chi, and Steven Dow

Introduction

Crowdsourcing involves recruiting large groups of people online to contribute small amounts of effort towards a larger goal. Increasingly, HCI researchers leverage online crowds to perform tasks, such as evaluating the quality of user generated content (Kittur, Suh, & Chi, 2008), identifying the best photograph in a set (Bernstein, Brandt, Miller, & Karger, 2011), transcribing text when optical character recognition (OCR) technologies fail (Bigham et al., 2010), and performing tasks for user studies (Heer & Bostock, 2010; Kittur, Chi, & Suh, 2008).

This chapter provides guidelines for how to use crowdsourcing in HCI research. We explore how HCI researchers are using crowdsourcing, provide a tutorial for people new to the field, discuss challenges and hints for doing crowdsourcing more effectively, and share three concrete case studies.

S. Egelman (✉)
Electrical Engineering & Computer Sciences, University of California, 731 Soda Hall,
Berkeley, CA 94720, USA
e-mail: egelman@cs.berkeley.edu

E.H. Chi
Google, Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA
e-mail: edchi@google.com

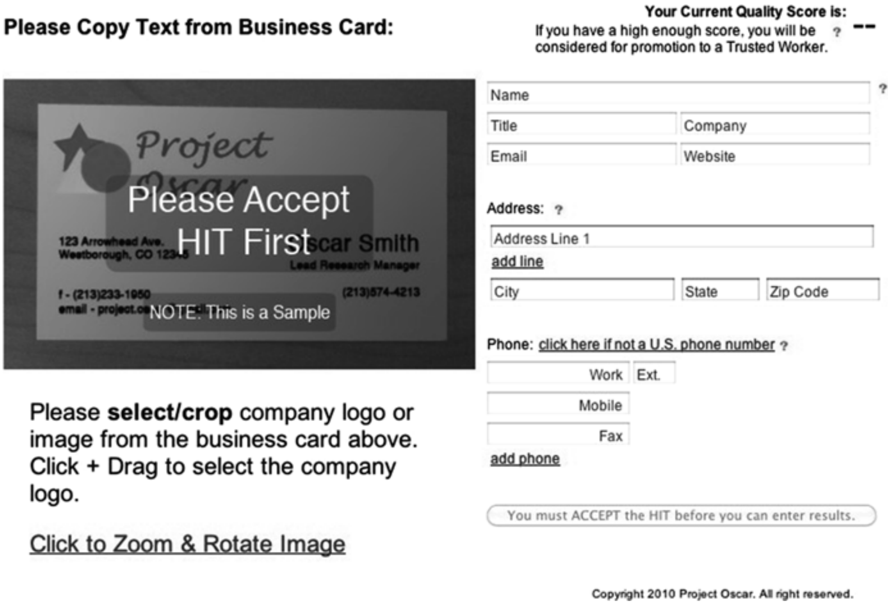
S. Dow
Human-Computer Interaction Institute, Carnegie Mellon University,
5000 Forbes Avenue, Newell Simon Hall, Pittsburgh, PA 15213, USA
e-mail: spdow@cs.cmu.edu

J.S. Olson and W.A. Kellogg (eds.), *Ways of Knowing in HCI*,
DOI 10.1007/978-1-4939-0378-8_11, © Springer Science+Business Media New York 2014

What Is Crowdsourcing?

Numerous online crowdsourcing platforms offer people micro-payments for completing tasks (Quinn & Bederson, 2011), but non-paid crowdsourcing platforms also exist. Non-paid crowd platforms typically offer some other value to users, such as embedding the task in a fun game (von Ahn & Dabbish, 2004) or engaging people in a cause, such as citizen science projects like Fold It, a protein folding effort (Hand, 2010). The increasing availability of crowdsourcing platforms has enabled HCI researchers to recruit large numbers of participants for user studies, to generate third-party content and quality assessments, and to build novel user experiences.

One canonical example of paid crowdsourcing from the crowdsourcing industry is business card data entry. Even very sophisticated algorithms utilizing OCR technology cannot deal with the great variety of different types of card designs in the real world. Instead, a company called CardMunch uploads business cards to Amazon's Mechanical Turk (MTurk)[1] to have them transcribed.[2] This way, a user who collects hundreds of business cards from a convention can have them transcribed very quickly and cheaply. Figure 1 shows the interface



Please Copy Text from Business Card:

Your Current Quality Score is: --
If you have a high enough score, you will be considered for promotion to a Trusted Worker.

Please Accept HIT First

NOTE: This is a Sample

Please **select/crop** company logo or image from the business card above. Click + Drag to select the company logo.

You must ACCEPT the HIT before you can enter results.

Copyright 2010 Project Oscar. All right reserved.

Fig. 1 Example Business Card task in Amazon Mechanical Turk. Screenshot used by permission from LinkedIn.com

[1] http://www.mturk.com/.

[2] http://www.readwriteweb.com/archives/linkedin_updates_cardmunch_iphone_app.php.

used for transcription. This example illustrates the role crowdsourcing has played in merging computational algorithms with human intelligence. That is, in places where algorithms fall short, online crowds can supplement them with human computation.

There are many other examples of crowdsourcing that do not involve financial payments. One example is the online encyclopedia, Wikipedia,[3] where hundreds of thousands of contributors author and edit articles. Similarly, the Tiramisu project relies on GPS traces and problem reports from commuters to generate real-time arrival time predictions for a transit system (Zimmerman et al., 2011).

Another example of unpaid crowdsourcing is the reCAPTCHA project (von Ahn, Maurer, McMillen, Abraham, & Blum, 2008) where millions of Internet users translate small strings of scrambled text, typically to gain access or open a new user account. The purpose is twofold. The system verifies that the user is human, not some kind of automated algorithm. Moreover, the project aims to digitize old out-of-print books by giving users one known word and one unknown word. Over time and across many users, the system learns the probability distribution of the unknown words and eventually translates entire book collections. These non-paid crowdsourcing platforms demonstrate the variety of incentive mechanisms available.

A Brief History

Before the invention of electronic computers, organizations employed teams of "human computers" to perform various mathematical calculations (Grier, 2005). Within the past decade, this notion of human computation has once again gained popularity due to not just an increase in online crowdsourcing platforms but also because researchers have become better able to understand the limitations of *machine* computation.

In HCI research literature, the pioneering work of von Ahn and Dabbish first explored using game mechanisms in the "ESP Game" to gather labels for images (von Ahn & Dabbish, 2004). Kittur, Chi, & Suh, 2008) suggested the use of MTurk for user studies.

Since these two early works, a growing community of HCI researchers has emerged to examine and utilize crowdsourcing in its many forms. This is evident by both the presence of large workshops at top HCI conferences, such as the ACM CHI conference, as well as new workshops and conferences dedicated entirely to crowdsourcing, such as HCOMP and Computational Intelligence.

HCI researchers have explored crowdsourcing by:

1. Studying crowd platforms for intellectual tasks, e.g., Wikipedia and social search.
2. Creating "crowdsensing" applications, e.g., CMU's Tiramisu (Zimmerman et al., 2011) or Minnesota's Cyclopath (Priedhorsky & Terveen, 2008).

[3] http://www.wikipedia.org/.

3. Designing "games with a purpose," e.g., CMU's ESP Game (von Ahn & Dabbish, 2004) or U Washington's PhotoCity (Tuite, Snavely, Hsiao, Smith, & Popović, 2010).
4. Utilizing micro-task platforms (e.g., MTurk) for a variety of activities, ranging from user study recruitment to judgment gathering.

While HCI research has much to gain from studying existing large-scale online communities (such as Twitter, Google+, Reddit, or Wikipedia) or building new crowd-based platforms (e.g., Zimmerman et al., 2011), this chapter aims to provide a useful resource for people new to the domain. Given the wide variety of research in this space, this chapter focuses primarily on how HCI researchers can leverage general-purpose crowdsourcing platforms, which are often used for completing micro-tasks. They provide easy access to scalable, on-demand, inexpensive labor and can be used for many kinds of HCI research.

In the rest of this chapter, we first look at how crowdsourcing can be applied to typical HCI activities, such as conducting participant studies and recruiting independent judges. Second, we provide a number of considerations and tips for using crowds, including a short tutorial on Amazon's Mechanical Turk. Third, we share three case studies that explore how each of the authors has personally used crowdsourcing in his research. Finally, we explore new HCI applications for crowdsourcing research and provide links to additional crowdsourcing resources.

How HCI Researchers Can Leverage Crowds

For many common HCI research activities, the scale, diversity, availability, and affordability of online crowds provide value. This section covers several of the traditional HCI research activities that benefit from utilizing general-purpose crowdsourcing platforms. We describe more advanced uses of crowdsourcing later.

Conducting online surveys: Crowdsourcing provides a wonderful recruiting tool for surveys and questionnaires, because the ability to reach large populations allows researchers to select for specific demographics, as well as recruit diverse samples, as discussed in detail later.

To better select samples of workers, a number of researchers have been using MTurk to learn more about crowd workers themselves (Quinn & Bederson, 2011). For example, Ross, Irani, Silberman, Zaldivar, and Tomlinson (2010) learned that over the last few years the demographics of MTurk workers have been shifting from primarily US workers, to a split between US and Indian workers. This shift is partly due to the fact that MTurk started allowing people to receive payments in Indian Rupees.

Conducting experiments: Crowdsourcing provides a cheap and quick way to recruit participants for user studies or experiments. An early example of this was Kittur, Suh, and Chi's use of MTurk to conduct a user study about Wikipedia article quality

(2008). Heer and Bostock (2010) were able to replicate and extend previous studies of graphical perception focused on spatial encoding and contrast. Heer and Bostock estimated that their crowdsourced studies resulted in cost savings at a factor of six (ibid). Similarly, Egelman and colleagues performed several experiments to examine Internet users' security behaviors (Christin, Egelman, Vidas, & Grossklags, 2011; Egelman et al., 2010; Komanduri et al., 2011).

These researchers leveraged the time and cost savings of online crowds to examine many more experimental conditions than would have been possible in a laboratory setting. For instance, a week of recruiting might result in 100 laboratory participants, who would need to be paid at least $10 each to participate in a 10-min experiment. The same experiment posted on a crowdsourcing platform might yield over 1,000 online participants when paid $1 each. Toomim, Kriplean, Pörtner, and Landay (2011) have used MTurk to compare different user interfaces. They proposed that task completion rates by MTurk workers provide a new measure of the utility of user interfaces. Specifically, they hypothesized that a more usable UI leads to more workers finishing a task and for less money. With thousands of workers conducting tasks with a range of different UIs, the researchers were able to measure the relative dropout rates based on the quality of the UI and the payment amount.

Training of machine-learning algorithms: Other researchers have been using online crowds to gather training data for novel uses of machine learning. For example, Kumar, Kim, and Klemmer (2009) sought to develop software that will transform web content to new designs. The researchers recruited MTurk workers to help them tune an algorithm that converts one website's Document Object Model (DOM) into another. In the task, online workers were given two websites and then for any particular design element on one page, they were asked to find the corresponding element in the second page. With enough of these judgments, the machine-learning algorithm can "learn" the structural patterns that map content across different designs.

Analyzing text or images: The ESP Game was one of the first and best examples of crowdsourcing, where online participants "labeled" images as a secondary effect of playing a game (von Ahn & Dabbish, 2004). The game shows two online players the same image. To earn points, the players have to simultaneously guess the same word or phrase without communicating. The side product of this game interaction provides descriptive language for the image (i.e., "tags"). Since then, HCI researchers have adopted crowdsourcing to analyze text and images for various research goals. A number of researchers have used crowds to analyze/categorize texts, such as blog threads, Wikipedia entries, and tweets (André, Bernstein, & Luther, 2012).

For analyzing images, one early well-known example is the NASA Click workers,[4] who were unpaid volunteers from all corners of the Web that used a website to help identify and classify craters on Mars. This was also one of the earliest citizen science projects.

[4] http://beamartian.jpl.nasa.gov/.

Another creative user study using crowds was an experiment on the effect of emotional priming on brainstorming processes. Lewis, Dontcheva, and Gerber (2011) first used MTurk workers to judge the emotional affect of a set of images. These ratings allowed the researchers to select one positive, one negative, and one neutral image as the independent variable for a brainstorming experiment. The researchers found that priming with both positive and negative images can lead to more original idea generation than neutral imagery.

Gathering subjective judgments: A number of researchers have leveraged crowdsourcing to gather subjective quality judgments on content. For example, Kittur, Chi, and Suh's evaluation of Wikipedia article quality showed that MTurk workers generated ratings that correlated highly with expert Wikipedians' evaluations (2008).

Utilizing the subjective judgments of crowds, Dow et al. paid online crowds to judge banner ads created by participants in a design experiment (2010). They then conducted an experiment on the design process to examine whether creating and receiving feedback on multiple designs in parallel—rather than simply iterating serially—affects design results and exploration. Participants came to the lab and created web banner ads, and the resulting designs were launched online at Amazon MTurk to collect relative performance metrics, such as the quality and diversity of the ad designs. The judgments of online workers showed that the parallel process resulted in more diverse explorations and produced higher quality outcomes than the serial process.

Considerations and Tips for Crowdsourcing

In this section, we discuss some of the questions that researchers should be prepared to answer when deciding whether to use crowdsourcing. Many decisions are involved regarding what types of tasks and how workers should go about completing them. For instance,

- Are the tasks well suited for crowdsourcing?
- If it is a user study, what are the tradeoffs between having participants perform the task online versus in a laboratory?
- How much should crowd workers earn for the task?
- How can researchers ensure good results from crowdsourcing?

Here we breakdown these key questions, discuss the challenges of using online crowds, and offer tips to help overcome those challenges.

Finally, we illustrate how to use one particular crowdsourcing platform, Amazon's Mechanical Turk, and give an overview of other crowdsourcing platforms.

When Is Crowdsourcing Appropriate?

Crowdsourcing typically enables researchers to acquire a large amount of user data for a low-cost with fast turn-around times. However, while crowdsourcing can be used for many different things, and there are a wide variety of different crowdsourcing platforms, not every research project is well suited for crowdsourcing. Researchers must consider task complexity, task subjectivity, and what information they can (or need to) infer about their users when deciding whether they can collect sufficient data through crowdsourcing.

As with any research project, the researcher should start by writing down the questions that she hopes to answer. Next, she must determine what data she needs in order to answer those questions. Finally, she must decide whether a crowdsourcing platform is able to yield that data and whether it can do so reliably with the desired demographic.

For instance, on the one hand, when conducting a very short opinion survey that collects responses from as many people as possible in a very short amount of time, MTurk or Google's Consumer Survey[5] might be the most appropriate platform, because these platforms focus on reaching large samples of the general public. On the other hand, if a project requires advanced skills, a platform that focuses on domain experts, like oDesk[6] or 99designs,[4] might be more appropriate.

Crowdsourcing should generally be used for tasks that can be performed online with minimal supervision. Tasks that require real-time individual feedback from the researcher may not be appropriate for crowdsourcing. However, these guidelines are nuanced. For instance, while MTurk itself does not support many advanced ways of communicating with users, there is nothing preventing a researcher from using MTurk to redirect users to a website she controls wherein she can support more interaction with the workers.

There really are no hard rules as to what sorts of projects might benefit from a crowdsourcing approach. New crowdsourcing platforms and methodologies continue to enable researchers to conduct online tasks that were previously thought to be unsuited to crowdsourcing.

What Are the Tradeoffs of Crowdsourcing?

Just because a researcher believes she *can* use crowdsourcing to complete a particular research project does not mean that she *should*. While crowdsourcing presents many advantages over traditional laboratory or field experiments in which the researcher is directly interacting with participants, it also has drawbacks that researchers need to take into account.

[5] http://www.google.com/insights/consumersurveys/.

[6] http://www.odesk.com/.

In a laboratory or field experiment where subjects meet with researchers face-to-face, they may feel additional motivation to provide quality results due to the supervision (i.e., the "Hawthorne effect;" Landsberger, 1958). This is one tradeoff when performing unsupervised tasks online. For instance, unless there are clear quality controls, users may feel free to "cheat." Users who cheat rarely do so out of malice, but instead do so out of laziness. This is basic economics: if the same reward can be achieved for doing less work, many users will do so. In many crowdsourcing platforms, the researcher ultimately gets to decide which users receive remuneration. Therefore the issue is not so much preventing or minimizing cheating, but instead including quality controls so that the researcher may detect it and then reject those responses. We discuss this in more detail later in this section.

Another detriment to using crowdsourcing in experiments is the unavailability of qualitative observations. Unless the researcher has invested time in creating an environment that allows for detailed observations as the user completes the task, there is little way of gathering observational data on the steps the user took while submitting a response. On the other hand, supervised laboratory and field experiments provide researchers with opportunities to ask users follow-up questions, such as why a particular action was performed. (See Looking Back: Retrospective Study Methods for HCI in this volume.)

Finally, a benefit of crowdsourcing is that the low cost allows researchers to iteratively improve their experimental designs. When performing a laboratory or field experiment, pilot experiments are usually run on only a handful of participants due to the time and cost involved, which means that the opportunity to identify and correct potential pitfalls is drastically reduced. With crowdsourcing, because the cost is usually orders of magnitude lower per user, there is no reason why a researcher cannot run iterative pilot experiments on relatively large samples. Likewise, researchers can use the low cost as part of a quality control strategy: if multiple workers complete the same task, outliers can be detected and removed.

Who Are the Crowd Workers?

Prior to the availability of crowdsourcing platforms, HCI research involving diverse samples of human subjects was often prohibitively expensive. Researchers commonly recruited locally, using only coworkers or students recruited nearby. These convenience samples, while heavily biased, have been accepted in the research community because alternatives were not readily available. Of course, all research subject samples suffer from a bias: they include only those who are willing to participate in research studies. However, the advent of crowdsourcing has shown that much more diverse participant pools can be readily accessible (Kittur, Chi, & Suh, 2008). The ability to recruit participants from around the world raises other concerns; chief among them is being able to describe the participant demographics. Or put more succinctly, *who are these workers*?

For some types of HCI research, in which the goal is not to generalize findings to larger populations, participant demographics may not matter. For example, for purely creative endeavors, such as collecting user-generated artwork or designs, the locations or education levels of participants may not be of concern. Likewise, when ground truth is readily verifiable, such as using crowdsourcing for translation or transcription, demographics also may not matter. However, when the goal is to yield knowledge that is generalizable to a large population, such as rating photographs for emotional content, knowing the demographic might be crucial to the study's ecological validity.

Crowdsourcing suffers from the same shortcomings as other survey methods that involve collecting self-reported demographic data; survey respondents often omit responses to certain demographic questions or outright lie. Likewise, all research methods suffer from potential biases because the people who participate were only those who both saw the recruitment notice and decided to participate. When users are recruited using traditional methods, such as from a specific geographical area or due to a common interest (e.g., online forums), some amount of information is immediately known about the sample. However, crowdsourcing changes all of this because users are likely to come from more diverse backgrounds. As a first step in identifying workers, a researcher may want to think about limiting her sample to specific geographic areas. For instance, some studies have shown that the demographics of US-based MTurk users are similar to the demographics of US-based Internet users as a whole, though the former are slightly younger and more educated (Ipeirotis, 2010a; Ross et al., 2010). If the ability to restrict users by location is unavailable on the platform the researcher wishes to use, then the geolocations of the users' IP addresses may be a reasonable proxy for user locations.

Other demographic information, such as education level, age, or gender, may be harder to reliably collect. If demographic information is necessary, users should be asked to self-report it. As with traditional methods that collect self-reported demographics, this information suffers from the same shortcomings (i.e., users might omit it or provide incorrect information). The trustworthiness of self-reported demographics varies by platform. Third party services, such as CrowdFlower,[7] compile user statistics so that requesters can rely on having more demographic information, as well as a user's history of completing previous tasks. The bottom line is that researchers should be aware of the potential to reach a diverse sample, and think about the type of worker they wish to reach.

How Much Should Crowdworkers Be Paid?

Some crowdsourcing platforms reward users with intangible benefits, such as access to special content, the enjoyment of playing a game, or simply the knowledge that they are contributing to a community. For instance, users contribute to Wikipedia in

[7] http://www.crowdflower.com/.

order to extend the quality of publicly available knowledge; reCAPTCHA users transcribe words in order to prove that they are not computer programs trying to gain access to a system. However, on some platforms, many users expect monetary incentives to participate. This raises the question, *how much should workers earn for work?*

Payment amounts can have profound effects on experimental results. Pay too little, and one risks not attracting enough workers or only attracting a very specific demographic (i.e., those willing to work for very little). Pay too much, and one may quickly exhaust the budget or turn away potential workers who incorrectly estimate too much work is involved. Of course, the proper payment amount is governed by many factors, and the most important are: community standards for the platform being used, the anticipated amount of time to complete the task, and the type of work involved.

Knowing the target demographic is crucial for determining payment amounts. For instance, soliciting logos from users of the crowdsourcing platform for designs, 99designs, is likely to cost two orders of magnitude more than soliciting logo ideas from MTurk users. However, users on 99designs are often professional designers, and therefore payments and rewards are commensurate with experience and expertise. Of course, when using MTurk, one will likely have to filter through many more low-quality answers, potentially negating any cost differential (i.e., a researcher may pay one designer $100 on 99designs, whereas it may take paying 100 workers each $1 or more on MTurk to yield an acceptable design).

For tasks that do not leverage skilled workers, the rule of thumb is to offer payment relatively close to the prevailing minimum wage. This of course is a loaded term, especially when talking about workers who are based all over the world. Without explicitly restricting one's workers to a particular geographic location or socioeconomic class, the payment amount will add a selection bias to the sample. For instance, Christin et al. (2011) found that for the same task, when they increased the payment from $0.01 to $1.00, participants from the developed world—as a proportion of total participants—increased significantly. The obvious explanation for this is that when the payment was too low, participants from the developed world did not believe it was worth their time.

Prior to deploying a new task to be crowdsourced, researchers should always run pilot experiments to get a good idea of how long it will take to complete the task. Some crowdsourcing platforms even provide "sandbox" features that allow tasks to be tested in the experimental environment for free while the researcher prepares to deploy them. In these environments, one can modify a task while viewing it from the worker's perspective. When the researcher has a good estimate for the task's time commitment, the researcher can spend a few minutes surveying tasks of similar complexity that others are offering to get a better understanding of the current market rates. If the budget allows, researchers may want to consider pricing their tasks slightly higher than other similar tasks (e.g., 30 cents if other similar tasks are paying 25 cents). This may help them to reach a larger audience by making their tasks stand out. Paying too much, on the other hand, usually attracts noisy answers from participants trying to earn quick money. Gaming the system for an economic advantage is always irresistible for some workers.

For a given price, the complexity of the task also has a profound impact on workers' willingness to perform it. Researchers have found that cognitive tasks involving creative or personal contributions tend to require higher payment amounts. For example, researchers will likely need to pay users more to spend 10 min writing unique product reviews than to spend 10 min answering multiple-choice surveys.

How to Ensure Quality Work?

Because crowdsourcing deals with potentially broad and diverse audiences, it is important to be able to minimize poor-quality responses. Barring that, tasks should be designed to make poor-quality responses immediately identifiable, so that they can be easily removed post hoc. Some of the techniques for doing this come from survey design best practices that have existed for decades, described in more detail below. (See chapter on "Survey Research in HCI," this volume.)

The easiest way to increase work quality is by preventing workers with bad reputations from participating. Some crowdsourcing platforms allow requesters—those posting tasks—to leave feedback about each of their workers. Other platforms provide worker statistics, such as the percentage of accepted tasks that were completed to the satisfaction of the requester. It is then possible for requesters to set a threshold so that only workers who have exceeded a certain approval rating are eligible to participate in their tasks. However, the quality of reputation systems varies greatly across different crowdsourcing platforms.

The most important, yet hardest way of increasing the quality of workers' work is by carefully crafting the language on task instructions. As a general rule, instructions need to be as specific as possible, while also being succinct. Because workers come from very diverse backgrounds and are performing tasks unsupervised, the tasks need to be worded to avoid misunderstandings and minimize follow-up clarifications. Researchers might want to tailor their instructions based on participants' estimated reading levels (e.g., Flesch-Kincaid readability score); however, it generally takes several rounds of piloting and iterative changes in order to finalize task descriptions. This type of hardening of experimental procedure is also common in laboratory experiments, but can be somewhat heightened in crowdsourcing experiments since the researcher cannot be in a room with the subjects to clarify any confusion.

A researcher may design the most straightforward task but still get a significant number of fraudulent responses. If it is easy for workers to submit irrelevant responses in order to receive a payment, many invariably will. Kittur, Chi, and Suh showed that the key is in designing the task so that fraudulent or low-quality responses can be easily detected using well-established survey design techniques (2008). The easiest way of doing this is by adding additional questions to the task in which the ground truth is known (also referred to as "gold standard" questions). For instance, to help determine whether users read the questions, one might ask "how many letters are there in the word 'dog'?" or even something as simple as "please select 'false'."

Another way of detecting fraudulent responses is by including questions that require open-ended responses, which demonstrate that the worker read and understood the question, rather than selected a correct answer by chance (Fowler, 1995). For example, on a survey that consists of multiple-choice questions, a researcher should think about replacing one with a text-box response so that she can assess workers' diligence to the task. The text box does two things. First, it discourages would-be cheaters by increasing the effort required to provide a fake response so that it is closer to the effort required to provide a legitimate response. Second, free-text questions make it much easier to detect blatantly fraudulent responses, because the responses are usually either gibberish or off-topic, whereas fraudulent responses to multiple-choice questions are hard to separate out from the legitimate responses.

Finally, one of the greatest advantages of crowdsourcing is that it is relatively tolerant of mistakes because tasks can be altered, modified, and reposted very easily. If a researcher finds that she is having a hard time achieving the sample sizes that she requires, she can simply increase the payment amount and try again. If she is not yielding the type of data that she requires, she may want to reword the task or add additional instructions or questions and try again. Making modifications and redeploying research studies has previously been viewed as highly time-consuming and costly. But with crowdsourcing, researchers can iterate more easily with their experimental designs.

A Tutorial for Using Amazon's Mechanical Turk

To give an example of a general crowdsourcing platform, we provide a short tutorial of Amazon's Mechanical Turk (MTurk), the largest and most well-known platform for leveraging crowds of people. While many platforms exist for specific types of tasks, MTurk is the most popular one for general-purpose crowdsourcing, because it essentially supports any task that can be completed from within a web browser by an Internet user. For this reason, it has become widely used for research tasks ranging from surveys and behavioral experiments to creative design explorations.

The Basics

Like most crowdsourcing platforms, MTurk relies on two types of users: *workers* and *requesters*. A worker is someone who uses the platform for the purpose of completing tasks, whereas the requester is the person who posts and pays for those tasks, known as "HITs" (Human Intelligence Tasks) in MTurk. For the purpose of this example, assume a researcher wishes to recruit users to complete an online survey. To do this, she will need to post her survey to MTurk.

When creating a new HIT on MTurk, a researcher needs to consider and specify the following variables:

- Payment amount for each valid response.
- Total number of responses to be collected.
- The number of times a worker may complete the HIT.
- Time allotted for each worker to complete the HIT.
- Time before the HIT expires (regardless of the number of completed assignments).
- Time before results are automatically approved (i.e., if the requester does not approve/reject individual results in time).
- Qualification requirements (e.g., approval rate and geographic location).

Requesters have the choice between using Amazon's web interface, which allows for the creation of very basic web forms with minimal logic using their graphical interface or using MTurk's API to implement more complex features (such as embedding externally hosted content). Using the API means that one can use a common programming language (C/C++, Java, Python, Perl, etc.) to automate the process of posting HITs, approving workers' responses, and then ultimately compensating the workers. This way, developers can access crowds through their own software, without having to manually post tasks using the MTurk website. For this tutorial, we will assume the researcher uses Amazon's web interface and each question of her survey is a basic HTML web form element.

Qualification Tasks

If a researcher wants to target a particular type of worker, the naïve approach would be to add screening questions to the survey and then remove all respondents who do not meet the requirements post hoc. Of course, this is very costly because it involves compensating everyone who completes the survey earnestly, even those who the researcher does not want to ultimately include in her dataset. As another way of targeting specific types of users, MTurk offers "qualification HITs."

A qualification HIT can be used to screen potential workers before they are allowed to participate in future and more complex HITs. For instance, if a requester is trying to survey workers who are in the market for a new car, she might create a very quick qualification HIT wherein workers are surveyed about planned upcoming purchases. This survey is likely to be very short and pay relatively little; she might ask ten questions about future purchases and compensate workers $0.05 for their time. Based on workers' responses to this screening survey, the requester can then give selected workers a "qualification," which is a custom variable associated with their profile indicating that they completed the screening survey satisfactorily and are then eligible for follow-up HITs.

Finally, the requester adds a requirement to their main task that workers need to pass the qualification to be eligible to participate. This "real" survey is likely to be much longer and compensate workers much more, but since some irrelevant respondents are ineligible, the money is more efficiently spent. Using this method, the researcher may create a standing pool of eligible participants that she can approach again and again in the future for subsequent research tasks, by creating a list of all the workers to whom she has granted the qualification.

Beyond Basic Surveys

MTurk includes all that one needs to deploy basic surveys that use standard HTML elements (e.g., forms, radio buttons), but what happens when one wants to add more advanced logic or dynamic embedded content? Luckily, requesters are not limited to working with the interface elements that MTurk supports; they can also redirect workers to their own websites to complete HITs. For instance, for the aforementioned survey about car buyers, imagine the researcher wants workers to use a Flash applet to design their dream cars. To do this, the researcher would create the Flash applet on her own website and then direct the workers to this website in one of two ways. The first way of doing this is to make the HIT an "external question," where the HIT is hosted outside of MTurk and will therefore appear in an embedded frame. She may design what appears in the HIT's *iframe* as she sees fit, so long as she ensures that all data she wishes to collect gets sent as HTTP POST variables to a particular MTurk submission URL.

Of course, the easier way of directing workers to a different website to complete a task is by including a link in the HIT (e.g., "click here to open the survey in a new window"). The problem with this method is that the researcher will need to map users who completed the survey to workers on MTurk. To address this problem, a shared secret is needed. To give an example:

1. A worker visits MTurk and accepts the HIT.
2. In the HIT, the worker opens a new window for the survey, hosted on a separate website.
3. Once the worker completes the survey, the last page displays a secret word that the worker must submit to MTurk to receive compensation.
4. When the researcher downloads the MTurk results, there is no way of determining whether workers actually took the survey because it was on a different website. However, because the MTurk HIT asked them to submit the secret word shown on the last page of the survey, all responses not containing this secret word can be rejected (because there is no evidence they completed the survey).

This method has one obvious flaw: workers may talk to one another. There are several very popular online forums for MTurk workers to discuss recently completed HITs,[8,9] so it would be trivial for one of them to reveal the secret word to other workers. One way around this is to create a unique—or reasonably unique—shared secret for each of the workers. For instance, some survey websites allow

[8] http://turkopticon.differenceengines.com/.

[9] http://forum.mturk.com/.

researchers to create random numbers to display at the end of an externally hosted survey. A researcher can then ask workers to enter this same number into MTurk. To verify the responses, it becomes a matter of just matching the numbers in order to identify which results to reject. Alternatively, a researcher can program shared secrets based on an algorithm that can be verified. For instance, the algorithm might print out a 6-digit random number that is also a multiple of 39; multiple submissions that include identical numbers are likely to have colluded, whereas the researcher can also make sure that a worker did not simply enter a 6-digit number at random.

Managing Results

As results are submitted, the researcher can download real-time data files formatted as comma separated values (CSV). In addition to whatever data is explicitly collected as part of the HIT, MTurk also includes information such as unique worker identifiers and timestamps.

Once a worker submits a HIT, the requester then needs to decide whether or not to accept the worker's result. The API allows requesters to write scripts to automatically download newly submitted responses and then automatically decide whether or not to approve them. Likewise, requesters may also manually visit the website to view newly submitted results. If a HIT is not adjudicated within the specified time interval, it is automatically approved. If the worker did not follow the HIT's instructions or the requester has good reason to believe that the response is fraudulent (e.g., incomprehensible language, failure to correctly answer "gold standard" questions), the requester may reject the HIT. When a requester rejects a HIT, the worker does not receive compensation. Since MTurk uses worker approval rates as proxies for reputation, rejection also hurts a worker's reputation and may prevent that worker from completing future HITs that set a reputation threshold.

Closing the HIT

Finally, once a sufficient number of responses have been collected, the researcher will want to prevent additional workers from completing the HIT, as well as pay the workers who have completed it satisfactorily. When she receives either the target number of responses or the time limit passes, the HIT is said to have "expired" (i.e., it is no longer available for additional workers to complete).[10] Once the HIT is expired, one must make sure that all of the workers who completed the task satisfactorily have been paid (otherwise they will be paid automatically, regardless of the quality of their responses), which can also be done either from the web interface or the API. If the work is not satisfactory, requesters have the option of specifying a reason for the rejection.

[10] If for some reason the researcher wishes to expire the HIT early, this is possible to do from both the web interface and the API. Likewise, HITs can also be extended using either method.

Case Studies

In this section, we briefly describe our own experiences using crowdsourcing in research. In particular, we aim to give an informal account of difficulties we encountered and how they were addressed.

Case Study 1: Assessing Quality on Wikipedia

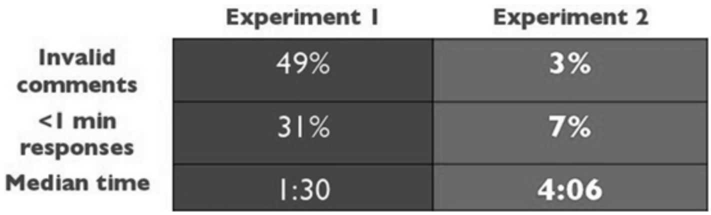
Ed H. Chi

In 2007, Aniket Kittur was an intern in my research group at PARC. One day early in the internship, we were exploring the question of how to assess the quality of Wikipedia articles. A huge debate was raging in the press about the quality of Wikipedia as compared with Encyclopedia Britannica articles (Giles, 2005). We became infatuated with the idea of using the crowd to assess the quality of the work of the crowd, and wanted to see if we could use Amazon MTurk to assess the quality of every Wikipedia article.

We knew that there was some limited ground truth data available on the quality of the articles from expert Wikipedians. In particular, one Wikipedia project systematically vetted a set of criteria for assessing the quality of Wikipedia articles, including metrics such as whether the articles were well-written, factually accurate, and used the required Neutral Point of View (NPOV). The project ranked some small set of articles with a letter grade from FA (Featured Article), A, GA (Good Article), B, C, and so on. By treating these ratings as ground truth, we embarked on a research project to find out if MTurk raters could reproduce expert ratings.

We asked workers to rate articles on a 7-point Likert scale on established metrics such as well-written, factually accurate, and of good quality. We also asked workers to give us free-text answers on how the articles could be improved. We paid workers $0.05 for each task. Within 2 days we had our data! Fifty-eight users made 210 ratings for 15 ratings per article, with a total cost of $10.50. We were thrilled!

However, the quality of the work was depressing. We obtained only a marginally significant correlation between the workers and the expert Wikipedians' consensus ratings ($r = 0.50$, $p = 0.07$). What was worse was that, by examining the rating data by hand, we saw that 59 % of the responses from workers appeared to be invalid. Forty-nine percent of the users did not enter any good suggestions on how to improve the articles, and 31 % of the responses were completed within 1 min, which is hardly enough time to actually read the article and form an opinion. What was worse was that 8 users appeared to have completed 75 % of the tasks! We felt frustrated and disappointed.



	Experiment 1	Experiment 2
Invalid comments	49%	3%
<1 min responses	31%	7%
Median time	1:30	4:06

Fig. 2 Dramatic Improvement in quality of the worker ratings on Wikipedia articles

We nearly gave up on the crowdsourcing approach at this point. We decided to try one more time. But in Experiment 2, we decided to completely change the design of the task:

- First, we decided that we would signal to the user that we were monitoring the results. We did this by asking some simple questions that were easy to answer just by glancing at the page. We used questions such as “How many images does this article have?” We could easily check these answers post hoc.
- Second, we decided to create questions where malicious answers were as hard to create as legitimate answers, such as “Provide 4–6 keywords that summarize this article.” These questions make it hard for a worker to “fake” reading the article. Not only did these questions require some cognitive processing, they also allowed us to see the types of tags that users would generate.
- Third, we made sure that answering the above questions was somewhat useful to completing the main task. That is, knowing how many sections or images the article had required the worker to pay some attention to whether the article was well-organized, which in turn was useful in making a decision about its quality.
- Fourth, we put the verifiable tasks ahead of the main task, so that the workers had to perform these steps before assessing the overall quality of the article.

To our surprise, the 2nd experiment worked much better, with 124 users providing 277 ratings for 20 ratings per article. We obtained a significant correlation with the Wikipedia ratings this time ($r = 0.66$, $p = 0.01$), and there was a much smaller proportion of malicious responses (3 % invalid comments, 7 % <1 min responses). Moreover, the time on task improved dramatically (4:06 min instead of 1:30 min)! We were happy with this success. More details can be found in our CHI2008 conference paper (Kittur, Chi, & Suh, 2008) (Fig. 2).

Case Study 2: Shepherding the Crowd

Steven Dow

When I was a postdoc in the HCI Group at Stanford, we started using crowdsourcing to enable our research. When we needed quality and similarity ratings on a set of visual designs for our experimental work on prototyping practices, we turned to online crowds from Mechanical Turk and oDesk.com (Dow et al., 2010). Through these experiences, we realized that platforms like MTurk offered no real opportunity to communicate with workers or to provide feedback that would help them improve their work performance. Along with Bjoern Hartmann, Anand Kulkarni, and Scott Klemmer, we built a system called Shepherd to understand the effects of introducing real-time feedback into a crowdsourcing platform (Dow, Kulkarni, Klemmer, & Hartmann, 2012).

Our goal was to get unskilled crowds to produce better results on complex work. While other research efforts take a computational approach to this problem and focus on workflows that sequence and coordinate small individual contributions (Bernstein et al., 2010; Kittur, Smus, Khamkar, & Kraut, 2011; Kulkarni, Can, & Hartmann, 2012; Little, Chilton, Goldman, & Miller, 2010a), our work on Shepherd took a more human-centered stance. If we want crowdsourcing to become a viable part of the economy, we cannot be satisfied with paying workers $2–3 per hour on average. Our work examined how we can make crowd work—and the people doing the work—more valuable.

It was our belief that we could educate and motivate workers to do more complex work through process improvements. In particular, we hypothesized that shepherding the crowd—by providing workers meaningful real-time feedback—would lead to better work, learning, and perseverance. We built the Shepherd system to inject feedback into the crowdsourcing process. In our task, the worker writes a series of reviews for products they own. As the reviews start piling in from multiple workers, a requester monitors a work dashboard, reviews each piece of work, and fills in a feedback form. Workers then receive this feedback before they start on their next product review. In the feedback form, workers see what they wrote previously, a checklist of effective product review strategies, and a Likert rating for the product review.

To understand the effects of external feedback on crowdsourcing performance, we conducted a between-subjects study with three conditions. Participants in the *None* condition received no immediate feedback, consistent with most current crowdsourcing practices. Participants in the *Self-assessment* condition judged their own work. Participants in the *External* assessment condition received expert feedback. We found that *Self-assessment* alone yielded better overall work than the *None* condition and helped workers improve over time. *External* assessment also yielded these benefits, but it also resulted in more work. Participants who received external assessment made more revisions to their original reviews. More details about the experimental setup and results can be found in our 2012 CSCW paper (Dow et al., 2012).

Case Study 3: Scaling Up Recruitment and Diversity

Serge Egelman

My research mostly focuses on how humans make decisions concerning privacy and security. This means that at least half of my time is spent conducting experiments on people, both in the laboratory and in the field. Prior to being introduced to crowdsourcing, large-scale online surveys were seen as highly laborious. Back when I was a graduate student at Carnegie Mellon, many researchers would use dedicated participant pools that largely consisted of students and staff. In order to yield more diverse demographics, my group generally shied away from these participant lists in favor of recruiting participants online.

We would post to online forums, such as Craigslist, and ask people to fill out our surveys in exchange for a raffle incentive (e.g., we would give away gift cards to randomly selected survey respondents). Posting recruitment notices became a full-time task. For instance, in the case of Craigslist, I would post to as many different cities as possible in order to get a diverse sample. This involved slightly changing the wording on the posting for each city to which I posted, since the Craigslist spam filter would flag similar-looking postings from different cities. This also involved keeping track of when postings were expiring and needed to be reposted. All this effort—2 weeks of graduate student time—generally resulted in about 100–200 responses per week.

Because of the large time investment, it was not feasible to modify experimental designs. That is, if our data prompted new questions that could only be addressed by adding additional material to a survey, our results could be delayed by several weeks.

It was not until late 2009 that I read an article comparing both the demographics and efficiency of MTurk workers with survey respondents who had been recruited by a market research firm (Jakobsson, 2009). Jakobsson found similar results between the two samples. This, in addition to reading articles by Ipeirotis on the demographics of MTurk workers (2008, 2010a, 2010b), led me to investigate whether I could use MTurk to recruit a diverse sample of survey respondents in a much shorter amount of time than previously possible.

In my first experiment, we recruited workers to complete a survey regarding their workplace file-sharing habits. We offered participants $0.25 to participate, and the survey took roughly 10–20 min. We received over 350 legitimate responses in the course of 48 h. Even more interestingly, over 95 % of our respondents held white-collar jobs and were completing our survey midday. This indicated that they were not "professional" experimental subjects…they were instead amusing themselves at work, rather than participating solely for the compensation. When contrasting our results with previous studies using other recruitment methods, we found that not only did crowdsourcing cost much less while enabling quicker recruitment, but the number of obvious "cheaters" (i.e., those who submitted nonsensical responses) had not increased.

Since then, crowdsourcing has become my go-to recruitment mechanism for experiments that can be completed online. In addition to surveys, this has also included interactive tasks using embedded applets (Egelman et al., 2010), as well as workers downloading custom software (Christin et al., 2011). To give another example, my colleagues and I used crowdsourcing to study password creation habits by recruiting over 5,000 participants (Komanduri et al., 2011). The cost per participant was roughly a dollar, while the quality of the results did not suffer—the paper received the honorable mention award at CHI 2011.

Prior to crowdsourcing, the thought that a researcher could recruit over 1,000 subjects in under a week for under $1,000 was unheard of, but this is the new reality.

Crowdsourcing Research and Resources

Beyond using existing general-purpose crowdsourcing platforms to serve core HCI research activities, a growing number of researchers are creating new crowdsourcing platforms—either from scratch or on top of existing general-purpose platforms—to explore novel systems and applications.

Crowd-powered software: The Soylent project is perhaps the best-known harbinger of using crowds as a first-class entity in a software application. Bernstein et al. (2010) created a word processing interface that enables writers to "hire" MTurk workers to shorten, proofread, and edit documents on demand. Soylent pioneered a *Find-Fix-Verify* pattern to help manage micro-task crowds by splitting tasks into a series of generation and review stages. Since this project appeared, a number of other crowd-powered systems have emerged including PlateMate, which uses crowd workers to perform nutrition analysis from photographs of food (Noronha, Hysen, Zhang, & Gajos, 2011).

Real-time crowdsourcing: One significant thrust by developers of crowd-powered systems has been the goal of tapping the abilities of crowds in (near) real-time. For example, to help answer everyday questions from visually impaired users, the VizWiz application asks the same question of multiple people at the same time through crowdsourcing (Bigham et al., 2010). To achieve near real-time response rates (just over 2 min on average), VizWiz proactivity recruited and queued workers to work on a simple separate task and then pulled them into the VizWiz application on an as-needed basis. A number of other real-time crowdsourcing applications have since emerged, including Adrenaline, which gets crowd workers to quickly filter a short video down to the best single photo (Bernstein et al., 2011), and Legion, which employs crowds to control UIs, such as a remote control interface for a robot (Lasecki, Murray, White, Miller, & Bigham, 2011).

Complex tasks with constraints: A key characteristic of crowdsourcing is the ability to employ people to make small contributions to a larger and more complex problem. Zhang et al. (2012) explored the use of crowds for trip itinerary planning, where a requester has specified any number of high-level goals and constraints (e.g., "at least one fresh local food restaurant"). The researchers created a collaborative planning system called Mobi that allows crowd workers to view the solution context and make additional changes based on current problem needs. This approach enables requesters to iteratively add, subtract, or re-prioritize goals; workers can contribute a small amount or continue working on the list of needs.

Crowd toolkits: Managing crowds can be challenging, especially for complex workflows. A number of research efforts focus on creating worker visualizations and workflow management tools. Kittur et al. (2011) implemented the CrowdForge workflow tool based on the MapReduce programming paradigm where tasks are partitioned into subproblems, mapped to workers, and then combined back into one result. Kulkarni et al. (2012) took a similar approach with Turkomatic by asking workers, "can you finish this task in 1 min? If not, please break the task down into multiple smaller tasks." The authors also developed workflow visualizations for Turkomatic to help requesters better facilitate this process.

Crowd-specific studies: Other crowdsourcing research focuses on gathering empirical data about how particular workflows and conditions affect the work performance and attitudes of crowd workers. For example, Little, Chilton, Goldman, and Miller (2010b) explored the tradeoffs of iterative and parallel processes for human computation tasks. They reported that, in general, iteration improves work quality, except on more generative tasks like brainstorming, where showing a previous worker's ideas may limit the creativity of the next worker. In addition to specific workflow issues, researchers have examined crowd feedback (Dow et al., 2012), social transparency (Stuart, Dabbish, Kiesler, Kinnaird, & Kang, 2012), and labor concerns (Quinn & Bederson, 2011) with respect to crowdsourcing environments.

Conclusions

Crowdsourcing offers a technique for recruiting lots of people online to perform work, which has the potential to change HCI research. By utilizing both paid workers and unpaid volunteers, researchers can greatly expand the diversity and reduce the time it takes to conduct user studies and large-scale data analysis. While this is a powerful method, the technique presents a number of potential pitfalls. This chapter summarizes these common pitfalls and gives examples of how to avoid them. We also included a short summary of how to use Amazon's Mechanical Turk so that researchers can quickly get started with this technique. However, this is a relatively new technique that continues to evolve rapidly. As such, we expect certain aspects of this chapter to become outdated in the future. It is our hope that HCI researchers will use the tips in this chapter to further refine and expand on this valuable new research method.

References

André, P., Bernstein M., & Luther K. (2012). Who gives a tweet?: Evaluating microblog content value. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (pp. 471–474). New York, NY: ACM

Bernstein, M. S., Brandt, J., Miller, R. C., & Karger, D. R. (2011). Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (pp. 33–42). New York, NY: ACM

Bernstein, M. S., Little G., Miller R. C., Hartmann B., Ackerman M. S., Karger D. R., et al. (2010). Soylent: A word processor with a crowd inside. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (pp. 313–322). New York, NY: ACM

Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., et al. (2010). VizWiz: Nearly real-time answers to visual questions. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (pp. 333–342). New York, NY: ACM

Christin, N., Egelman, S., Vidas, T., & Grossklags, J. (2011). It's all about the Benjamins: An empirical study on incentivizing users to ignore security advice. *Financial Cryptography and Data Security* 16–30

Dow, S. P., Glassco, A., Kass, J., Schwarz, M., Schwartz, D. L., & Klemmer, S. R. (2010). Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Transactions on Computer-Human Interaction (TOCHI), 17*(4), 18.

Dow, S., Kulkarni, A., Klemmer, S., & Hartmann, B. (2012). Shepherding the Crowd Yields better work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (pp. 1013–1022). New York, NY: ACM

Egelman, S., Molnar, D., Christin, N., Acquisti, A., Herley, C., & Krishnamurthi, S. (2010). Please continue to hold: An empirical study on user tolerance of security delays. In *Proceedings (Online) of the 9th Workshop on Economics of Information Security*

Fowler, F. J., Jr. (1995). *Improving survey questions: Design and evaluation* (Vol. 38). Thousand Oaks, CA: Sage. Incorporated.

Giles, J. (2005). Internet encyclopaedias go head to head. *Nature, 438*(7070), 900–901.

Grier, D. A. (2005). *When computers were human* (Vol. 316). Princeton, NJ: Princeton University Press.

Hand, E. (2010). Citizen science: People power. *Nature, 466*(7307), 685.

Heer, J., & Bostock, M. (2010). Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems* (pp. 203–212). New York, NY: ACM

Ipeirotis, P. (2008). *Mechanical turk: Demographics*. Retrieved September 15, 2009, from http://behind-the-enemy-lines.blogspot.com/2008/03/mechanical-turk-demographics.html

Ipeirotis, P. (2010a). Demographics of mechanical turk. Working Paper, CeDER-10-01. http://archive.nyu.edu/handle/2451/29585

Ipeirotis, P. (2010b). *The new demographics of mechanical turk*. Retrieved July 2, 2012, from http://www.behind-the-enemy-lines.com/2010/03/new-demographics-of-mechanical-turk.html

Jakobsson, M. (2009). Experimenting on mechanical turk: 5 How tos. Retrieved November 4, 2009, from http://blogs.parc.com/blog/2009/07/experimenting-on-mechanical-turk-5-how-tos/

Kittur, A., Chi, E. H., & Suh, B. (2008). Crowdsourcing user studies with mechanical turk. In *Proceedings of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems* (pp. 453–456). New York, NY: ACM

Kittur, A., Smus, B., Khamkar, S., & Kraut, R. E. (2011). Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (pp. 43–52). New York, NY: ACM

Kittur, A., Suh, B., & Chi, E. H. (2008). Can you ever trust a Wiki?: Impacting perceived trustworthiness in Wikipedia. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work* (pp. 477–480). New York, NY: ACM

Komanduri, S., Shay, R., Kelley, P. G., Mazurek, M. L., Bauer, L., Christin, N., et al. (2011). Of passwords and people: Measuring the effect of password-composition policies. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems* (pp. 2595–2604). New York, NY: ACM

Kulkarni, A., Can, M., & Hartmann, B. (2012). Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (pp. 1003–1012). New York, NY: ACM

Kumar, R., Kim, J., & Klemmer, S. R. (2009). Automatic retargeting of web page content. In *Proceedings of the 27th International Conference (Extended Abstracts) on Human Factors in Computing Systems* (pp. 4237–4242). New York, NY: ACM

Landsberger, H. A. (1958). *Hawthorne revisited: Management and the worker, its critics, and developments in human relations in industry*. Ithaca, NY: Cornell University.

Lasecki, W. S., Murray, K. I., White, S., Miller, R. C., & Bigham, J. P. (2011). Real-time crowd control of existing interfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (pp. 23–32). New York, NY: ACM

Lewis, S., Dontcheva, M., & Gerber, E. (2011). Affective computational priming and creativity. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems* (pp. 735–744). New York, NY: ACM

Little, G., Chilton, L. B., Goldman, M., & Miller, R. C. (2010a). TurKit: Human computation algorithms on mechanical turk. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (pp. 57–66). New York, NY: ACM

Little, G., Chilton, L. B., Goldman, M., & Miller, R. C. (2010b). Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD Workshop on Human Computation* (pp. 68–76). New York, NY: ACM

Noronha, J., Hysen, E., Zhang, H., & Gajos, K. Z. (2011). Platemate: Crowdsourcing nutritional analysis from food photographs. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (pp. 1–12). New York, NY: ACM

Priedhorsky, R., & Terveen, L. (2008). The computational geowiki: What, why, and how. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work* (pp. 267–276). New York, NY: ACM

Quinn, A. J., & Bederson, B. B. (2011). Human computation: A survey and taxonomy of a growing field. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems* (pp. 1403–1412). New York, NY: ACM

Ross, J., Irani, L., Silberman, M., Zaldivar, A., & Tomlinson, B. (2010). Who are the crowdworkers?: Shifting demographics in mechanical turk. In *Proceedings of the 28th International Conference (Extended Abstracts) on Human Factors in Computing Systems* (pp. 2863–2872). New York, NY: ACM

Stuart, H. C., Dabbish, L., Kiesler, S., Kinnaird, P., & Kang, R. (2012). Social transparency in networked information exchange: A theoretical framework. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (pp. 451–460). New York, NY: ACM

Toomim, M., Kriplean, T., Pörtner, C., & Landay, J. (2011). Utility of human-computer interactions: Toward a science of preference measurement. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems* (pp. 2275–2284). New York, NY: ACM

Tuite, K., Snavely, N., Hsiao, D. -Y., Smith, A. M., & Popović, Z. (2010). Reconstructing the world in 3D: Bringing games with a purpose outdoors. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games* (pp. 232–239). New York, NY: ACM

von Ahn, L., & Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 319–326). New York: ACM

von Ahn, L., Maurer, B., McMillen, C., Abraham, D., & Blum, M. (2008). reCAPTCHA: Human-based character recognition via web security measures. *Science, 321*(5895), 1465–1468.

Zhang, H., Law, E., Miller, R., Gajos, K., Parkes, D., & Horvitz, E. (2012). Human computation tasks with global constraints. In *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems* (pp. 217–226). New York, NY: ACM

Zimmerman, J., Tomasic, A., Garrod, C., Yoo, D., Hiruncharoenvate, C., Aziz, R., et al. (2011). Field trial of Tiramisu: Crowd-sourcing bus arrival times to spur co-design. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems* (pp. 1677–1686). New York, NY: ACM