2014

Human-Computer Interaction and Collective Intelligence

Jeffrey P. Bigham
Carnegie Mellon University

Michael S. Bernstein
Stanford University

Eytan Adar
University of Michigan

Published In
The Collective Intelligence Handbook, Thomas W. Malone and Michael S. Bernstein (Editors).

Human-Computer Interaction and Collective Intelligence

JEFFREY P. BIGHAM
Carnegie Mellon University
MICHAEL S. BERNSTEIN
Stanford University
and EYTAN ADAR
Univerity of Michigan

1. INTRODUCTION

Human-computer interaction (HCI) works to understand and to design interactions between people and machines. Increasingly, human collectives are using technology to gather together and coordinate. This mediation occurs through volunteer and interest-based communities on the web, through paid online marketplaces, and through mobile devices.

The lessons of HCI can therefore be brought to bear on different aspects of collective intelligence. On the one hand, the people in the collective (*the crowd*) will only contribute if there are proper incentives and if the interface guides them in usable and meaningful ways. On the other, those interested in leveraging the collective need usable ways of coordinating, making sense of, and extracting value from the collective work that is being done, often on their behalf. Ultimately, collective intelligence involves the co-design of technical infrastructure and human-human interaction: a socio-technical system.

In crowdsourcing, we might differentiate between two broad classes of users: *requesters* and *crowd members*. The requesters are the individuals or group for whom work is done or who takes the responsibility to aggregate the work done by the collective. The crowd member (or crowd worker) is one of many people to contribute. While we often use the word "worker," crowd workers do not have need to be (and often aren't) contributing as part of what we might consider standard "work." They may work for pay or not, work for small periods of time or contribute for days to a project they care about, and they may work in such a way as each individual's contribution may be difficult to discern from the collective final output.

HCI has a long history of studying not only the interaction between individuals with technology, but also the interaction of groups with or mediated by technology. For example, computer-supported cooperative work (CSCW) investigates how to allow groups to accomplish tasks together using a shared or distributed computer interfaces, either at the same time or asynchronously. Current crowdsourcing research alters some of the standard assumptions about the size, composition, and stability of these groups, but the fundamental approaches remain the same. For instance, workers drawn from the crowd may be less reliable than groups of employees working on a shared task, and group membership in the crowd may change more quickly.

There are three main vectors of study for HCI and collective intelligence. The first is *directed crowdsourcing*, where a single individual attempts to recruit and guide a large set of people to help accomplish a goal. The second is *collaborative crowdsourcing*, where a group gathers based on shared interest and self-determine their organization and work. The third vector is *passive crowdsourcing*, where the crowd or collective may never meet or coordinate, but it is still possible to mine their collective behavior patterns for information. We cover each vector in turn. We conclude with a list of challenges for researches in HCI related to crowdsourcing and collective intelligence.

2. DIRECTED CROWDSOURCING

Directed crowdsourcing describes systems where an algorithm or person directs workers to pursue a specific goal. For example, a requester might seek to gather a crowd to tag images with labels or to translate a poem. Typically, this involves a single requester taking a strong hand in designing the process for the rest of the crowd to follow.

This section overviews work in directed crowdsourcing, including considerations to be made when deciding whether a particular problem is amenable to directed crowdsourcing, how to design tasks, and how to recruit and involve workers.

Workers in directed crowdsourcing generally complete tasks that the requester asks to be completed. Why would they perform the task? Sometimes the goals of requesters and workers are aligned, as is the case in much of the crowdsourcing work being done in citizen science. For instance, the crowd cares about a cause, e.g. tracking and counting birds [Louv et al. 2012], and the requester's direction is aimed primarily at coordinating and synthesizing the work of willing volunteers.

In other situations, crowd workers may not share the same goal as the requester. In this case, one challenge is to design incentives that encourage them to participate. This can be tricky because workers may have different reasons for participating in crowd work: for example, money, fun, or learning.

Several systems have introduced elements of games into crowd tasks, i.e. *gamified* them, to incentivize workers by making the tasks more enjoyable. For instance, the ESP Game is a social game that encourages players to label images by pairing them with a partner who is also trying to label the same image [Von Ahn and Dabbish 2004]. Likewise, FoldIt players found stable protien configurations that had eluded scientists for a decade [Khatib et al. 2011]. A challenge with gamification is that it can take years and significant insight to convert most tasks to games that people will want to play. It can also be difficult to attract players to the game, and the games popularity may change over time. Some of these games have been successful, while many others have attracted few players.

Another option is to pay crowd workers. Paid crowdsourcing differs from traditional contract labor in that it often occurs at very small timescales (for similarly small increments of money), and often interaction with the worker can be fully or partially programmatic. This form of crowdsourcing often takes place in online marketplaces such as Amazon Mechanical Turk and Elance-oDesk. In paid crowdsourcing, a workers incentive is ostensibly money, although money is not the only motivator even in paid marketplaces [Antin and Shaw 2012]. Money may affect but cannot be reliably used to improve desirable features of the game play, e.g. quality of the work, timeliness of the work [Mason and Watts 2010]. Because of the ease by which paid workers can be recruited, paid crowdsourcing, especially Amazon Mechanical Turk, is a popular prototyping platform for research and product in crowdsourcing.

An alternative approach is to collect crowdsourcing as a result (or by-product) of something else the user (or worker) wanted to do. For instance, reCAPTCHA is popular service that attempts to differentiate people from machines on the web by presenting a puzzle comprised of blurry text that must be deciphered in order to prove that one is human [von Ahn et al. 2008]. As opposed to other CAPTCHAs, reCAPTCHA has a secondary goal of converting poorly-scanned books to digital text. reCAPTCHA presents two strings of text, one that it knows the answer to and one that it does not. By typing the one it knows the answer to, the user authenticates himself. By typing the one it does not know, the user contributes to the goal of digitizing books. DuoLingo uses a similar approach to translate documents on the web into new languages as a byproduct of users learning a foreign language [Hacker and von Ahn 2012].

2.1. Quality & Workflows

When a requester asks for people to help her with a task, they will often do exactly what is asked, but not quite what was desired. Quality control mechanisms and workflows can help ensure better results. Interestingly, however, the quality of outputs from paid crowdsourcing is not a clear function of the price paid. For instance, paying more has been found to increase the quantity but not necessarily the quality of the work that is done [Mason and Watts 2010]. The usability of the task at hand can affect how well workers perform at the task. Human-computer interaction offers crowdsourcing methods for engineering tasks that crowd workers are likely to be able to do well with little training. Because crowd workers are often assumed to be new to any particular task, designing to optimize learnability is important, whereas other usability dimensions like efficiency or memorability may be less so.

Crowdsourcing tasks are often decomposed into small, atomic bits of work called microtasks. As a result, workers may not understand how their contribution fits into a broader goal and this can impact the quality of their work. One approach for compensating for the variable quality of the work received and for combining the small efforts of many workers is to use a workflow, also sometimes called a crowd algorithm. Some common workflows are iterative improvement [Little et al. 2009], parallel work followed by a vote [Little et al. 2009], map-reduce [Kittur et al. 2011], find-fix-verify (FFV) [Bernstein et al. 2010a], and crowd clustering [Chilton et al. 2013]. Good workflows help to achieve results that approach the performance of the average worker in the crowd, and sometimes can help achieve the ”wisdom of the crowd” effect of the group being better than any one individual. Practically, they also allow large tasks to be consistently completed, even if each worker only works on the task for a short amount of time.

For example, Soylent is a Microsoft Word plugin that allows the crowd to help edit documents, for instance fixing spelling/grammar, or shortening the document without changing its meaning. It introduced the Find-Fix-Verify workflow, which proceeds in 3 steps: (i) workers find areas in the document can could be appropriate for improvement, (ii) a second set of workers propose candidate changes (fixes), and (iii) a third set of workers verify that the candidate changes would indeed be good changes to make. The FFV workflow has a number of benefits. First, it was observed previously that workers tended to make the least acceptable change. For instance, if they were asked to directly fix the document or to make it smaller they would find a single change to make. The “find” step encourages multiple workers to find many places to fix in the document (or their assigned chunk of the document). The fix and verify steps are then scoped to that particular location. It was observed that resulted in more problems being found and fixed.

Workflows can often get complex, requiring many layers of both human and machine interaction. For instance, PlateMate combines several crowd-powered steps with machine-powered steps into a complex workflow that is able to match the performance of expert dieticians in determining the nutritional content of a plate of food [Noronha et al. 2011]. For a new problem that one wants to solve with crowdsourcing a challenge can be coming up with an appropriate workflow that allows crowd workers to contribute toward the end goal.

As crowdsourcing broadens from amateur microtasks to goals involving groups of interdependent experts, the nature of these workflows may evolve. Flash Teams offer one vision, where computation acts as a coordinating agent to draw together diverse experts from platforms such as Elance-oDesk [Retelny et al. 2014]. These kinds of approaches have already enabled crowdsourcing of a broad class of goals including design prototyping, course development, and film animation, in half the work time of traditional self-managed teams.

2.2. Interactive Crowd-Powered Systems

Traditional workflows can be quite time-consuming as each iteration requires crowd workers to be recruited and to perform their work. Near real-time workflows use time as a constraint and often work by having workers work in parallel and then having either an automatic or machine process make sense of the work as it is produced. The first step is to pre-recruit a group of workers who are then available to do work at interactive speeds once the work to be done is available. The SeaWeed system pre-recruited a group of workers who would then collectively play economics games [Chilton et al. 2009]. VizWiz pre-recruited workers and had them answer old questions until a new question came in for them [Bigham et al. 2010]. Adrenaline used a retainer pool to recruit a group of workers and then showed that this group could be called back quickly [Bernstein et al. 2011]. Workers in the retainer are paid a small bonus to be part of the pool, and collect these earnings if they respond quickly enough when asked. Turkomatic recruits workers and then lets them be programmatically sent to a task as a group [Kulkarni et al. 2011].

There is also value in getting the workers to work together synchronously. One reason to do this is to build real-time systems that are able to compensate for common problems in the crowd — namely, workers sometimes perform poorly and sometimes leave the task for something else to do. For instance, the Legion system puts crowd workers in control of a desktop interface by having them all contribute keyboard and mouse commands [Lasecki et al. 2011]. The crowd worker who for a given time interval is most similar to the others is elected a leader to assume full control, thus balancing the wisdom of the crowds with a real-time time constraint. This system was used across a variety of desktop computing applications and even to control a wireless robot. Adrenaline uses a similar concept to quickly refine and then eventually pick a high-quality frame from a digital video, thus creating a real-time crowd-powered camera.

Another reason to work as a group is to accomplish a goal that no worker could accomplish alone. The Scribe system allows a group of workers to collectively type part of what they hear in real-time along with a speaker [Lasecki et al. 2012]. An automated process then stitches the pieces back together using a variant of Multiple Sequence Alignment [Naim et al. 2013]. No worker is able to keep up with natural speaking rates alone, but collectively they can by using this approach. Employing a group allows the task to be made easier in ways that would not be possible if a single person were responsible for typing everything. Most obviously, each worker only has to type part of what he hears, but, more interestingly, when working as part of a group each workers task can be made even easier. The audio of the portion of speech the workers is expected to type can be algorithmically slowed down, which allows the worker to more easily keep up [Lasecki et al. 2013]. The remainder of the audio is then sped up so that the worker can keep context. Overall, this increases recall and precision, and reduces latency.

2.3. Programming with the Crowd

Crowd-powered systems behave differently than completely automated systems, and a number of programming environments have been constructed to assist designing and engineering them. For instance, crowd workers are often slow and expensive, so TurkIt allows programs to reuse results from tasks sent to the crash, employing a “crash and run” paradigm that allows for easier programming [Little et al. 2009]. Both VizWiz and Soylent, for example, were programmed using TurKit as a scaffold. AskSheet embeds crowd work into a spreadsheet and helps to limit steps that need to be done by crowd workers in order to make decisions [Quinn and Bederson 2014].

Jabberwocky exposes workflows as programming language primitives, and supports operation on top of a number of different kinds of crowds, including both Mechanical Turk and also social sources like Facebook [Ahmad et al. 2011]. One of the workflows it makes easily available is a crowdsourcing equivalent of Map Reduce called Man Reduce. Jabberworky builds on CrowdForge's approach for having work automatically divided up in the Map step for multiple workers to each complete, and then combined back together in the reduce step [Kittur et al. 2011]. One example of this is writing an essay by assigning different paragraphs to different workers and then having a reduce step in which those paragraphs are combined back together.

When crowd-powered systems do not behave as expected, it can be difficult to figure out why. Some systems have been developed to allow for the equivalent of debugging. For instance, CrowdScape records low-level features of how crowd workers perform their task and then allows requesters to easily visualize these recordings [Rzeszotarski and Kittur 2012]. This can help to identify confusing aspects of the task, understand where improvements are most needed, e.g. in code or the crowd tasks, and allow requesters to understand performance even on subjective tasks. The scrolling, key presses, mouse clicks, etc. that collectively define a *task fingerprint* can be useful in understanding how the work was done. If a user was requested to read a long passage and then answer a question about it, we might assume that the work was not done well if they scrolled quickly past the text and immediately input an answer.

2.4. Drawbacks and ethics of microtasking

The human-computer interaction field is acutely aware of the effects that the sociotechnical systems that are created may have on the future of crowd work. Crowds bear the potential of mass action and people power. Yet as Irani and Silberman have argued [Irani and Silberman 2013], AMTs design directs this collective power into reliable, steadily humming computational infrastructure. This infrastructure is designed to keep questions of ethical labor relations or worker variation out of requesters (employers) sight. This may lead to undesirable consequences for crowdsourcing, such as workers being divorced from the tasks they work on and reduced value being assigned to expertise. Microtasks remain popular despite their drawbacks because they are accessible and often low-effort for requesters.

One way that crowd work has been viewed in the past is as a source of very low-cost labor. Because this labor sometimes provides low-quality input, techniques need to be derived to compensate for it. One of the goals of human-computer interaction research in crowdsourcing is to demonstrate the potential for a brighter future for crowd work in which workers are able to accomplish together something that they could not have accomplish on their own.

It may be tempting in crowd work to treat workers as program code. Some have recognized that this prevents many of the benefits of crowd workers being human from being realized. For instance, once crowd workers learn a new task, they are likely to be better (faster, more accurate) at it. As a result, it may make sense to keep a worker around over time completing similar work to improve throughput, which is advantageous to both workers and requesters.

Concerns about labor practices have led to work exploring current demographics of workers and work that explicitly considers how to improve working conditions. "The Future of Crowd Work" notes a number of suggestions for improving crowd work, including allowing workers to learn and acquire skills from participating in crowd work [Kittur et al. 2013]. A common but incorrect notion about Mechanical Turk, for instance, is that workers are mostly anonymous — however, this has since been shown not to be true [Lease et al. 2013]. A growing theme among human-computer interaction research is in realizing *both* the advantages of a programmatically-available source of human intelligence, and the essential humanness of the participants.

3. COLLABORATIVE CROWDSOURCING

Many of the most famous crowdsourcing results are not directed. Instead, they depend on volunteerism or other non-monetary incentives for participation. For example, volunteer crowds have:

— Authored Wikipedia[1], the largest encyclopedia in history,
— Helped NASA identify craters on Mars [Kanefsky et al. 2001],
— Surveyed satellite photos for images of a missing person [Hellerstein and Tennenhouse 2011],
— Held their own in chess against a world champion [Nalimov et al. 1999],
— Solved open mathematics problems [Cranshaw and Kittur 2011b],
— Generated large datasets for object recognition [Russell et al. 2008],
— Collected eyewitness reports during crises and violent government crackdowns [Okolloh 2009], and
— Generated a large database of common-sense information [Singh et al. 2002].

Each of these successes relied on the individuals' intrinsic motivation to participate in the task. *Intrinsic motivation*, unlike extrinsic motivators from the previous section such as money, means that participants bring their own excitement to the table: for example, via interest in the topic, desire to learn, or drive to experience new things [Ryan and Deci 2000].

While these systems often have leaders, as in Jimmy Wales' leadership of Wikipedia or the main contributors to an open source project, we refer to these systems as *collaborative crowdsourcing* efforts here. Doing so distinguishes them from the earlier efforts where a single individual is more directly driving the group's vision and tasks. Often, in collaborative crowdsourcing, the group exercises self-determination and self-management to plan its own future before executing it.

Human-computer interaction research first seeks to understand these sociotechnical systems. Why do they work? What do they reveal about the human processes behind collective intelligence? How do changes to the design or tools influence those processes?

Second, in parallel, human-computer interaction research aims to empower these self-directed systems through new designs. These designs may be minor changes that produce large emergent effects, for example recruiting more users to share movie ratings [Beenen et al. 2004]. Or, they may be entirely new systems, for example creating a community to capture 3-D models of popular locations through photographs [Tuite et al. 2011].

This research plays itself out across themes such as leadership, coordination, and conflict as well as through the type of information provided by the crowd. For each aspect, many decisions will influence the effectiveness and fit of a particular design for different contexts. Here, we look at each in turn.

3.1. Leadership and decision-making

When the group is self-organizing, decision-making becomes a pivotal activity. Does the group spend more time debating its course of action than actually making progress?

Niki Kittur and colleagues undertook one the most well-known explorations of this question, using Wikipedia as a lens [Kittur et al. 2007]. The authors obtained a com-

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

plete history of all Wikipedia edits, then observed the percentage of edits that were producing new knowledge (e.g., edits to article pages) vs. edits that were about coordinating editing activities (e.g., edits to talk pages or policy pages). Over time, the number of article edits decreased from roughly 95% of activity to just over half of activity on Wikipedia. The result suggests that as collective intelligence activities grow in scope and mature, they may face increased coordination costs.

Leadership faces other challenges. Follow-on work discovered that as the number of editors on an article increases, the article's quality only increases if the editors take direct action via edits rather than spend all their effort debating in Wikipedia's talk pages [Kittur and Kraut 2008]. In policy decisions, senior members of the community have more than average ability to kill a proposal, but no more than average ability to make a proposal succeed [Keegan and Gergle 2010]. In volunteer communities, it may be necessary to pursue strategies of distributed leadership [Luther et al. 2013]. These strategies can be more robust to team members flaking unexpectedly.

In terms of design, socializing new leaders is challenging. There is a debate whether future leaders are different from even their earliest activities [Panciera et al. 2009] or whether they are just like other members and can be scaffolded up through legitimate peripheral participation [Lave and Wenger 1991; Preece and Shneiderman 2009]. Software tools to help train leaders can make the process more successful [Morgan et al. 2013].

3.2. Coordination

Crowds can undertake substantial coordination tasks on demand. In these situations, emergent coordination among crowds takes the place of explicit leadership. Crisis informatics, as one example, focuses on coordination in the wake of major disasters such as earthquakes and floods. Groups have adopted social media such as Twitter to promote increased situational awareness during such crises [Vieweg et al. 2010]. When official information is scarce and delayed, affected individuals can ask questions and share on-the-ground information; remote individuals can help filter and guide that information so it is most impactful [Starbird 2013].

Coordination can be delicate. On one hand, successful scientific coordinations such as the Polymath project demonstrate that loosely guided collaborations can succeed [Cranshaw and Kittur 2011a]. In the Polymath project, leading mathematicians blogged the process of solving a mathematics problem and recruited ideas and proofs from their readers. On the other hand, distributed voting sites such as Reddit exhibit widespread underprovision of attention [Gilbert 2013] — in other words, the users' attention is so focused on a few items that they often miss highly viral content the first time it is posted. Platforms such as Kickstarter may offer a "living laboratory" [Chi 2009] for studying collective coordination efforts at large (e.g., [Gerber et al. 2012]).

3.3. Conflict

Most collective intelligence systems will produce internal conflict. Some systems are even *designed* to host conflict (e.g., systems to support deliberative democracy). The objective of such designs is not to remove or resolve disagreement, but rather to show participants points of agreement and contention.

For example, Reflect [Kriplean et al. 2011b] and ConsiderIt [Kriplean et al. 2011a] are designed to host discussion and debate. In order to do so, they introduce procedural changes into the format of the discussion. For example, Reflect asks each commenter to first summarize the original poster's points. ConsiderIt, focused on state election propositions, instead asks visitors to author pro/con points rather than leave unstructured comments.

Design may also aim to increase awareness of other perspectives. By visualizing how much users' content consumption is biased, browser plugins can encourage readers to balance the news perspectives [Munson et al. 2013]. Projects such as OpinionSpace [Faridani et al. 2010] and Widescope [Burbank et al. 2011] likewise demonstrate how even people who disagree on binary choices may in practice be closer in opinion than they think.

3.4. Participation

Though leadership, cooperation, and the management of conflict are important in the construction of effective crowd systems, continuous, active participation is critical. Without participation in collective intelligence activities, there is no collective, and thus no intelligence. The online communities literature has devoted considerable energy to studying how to attract and maintain participation. Ideas of incentives and motivation are also extensively discussed in Chapters 5 and 8.

The GroupLens project has produced some of the most influential research investigating this question. Years ago, GroupLens created the MovieLens site, which was an early movie recommender service, and it attracted a steady amount of volunteer usage. The researchers then began applying concepts from social psychology to increase participation on MovieLens. For example, they found that calling out the uniqueness of a user's contributions and creating challenging but achievable goals increased the number of movies that users would rate on the site [Beenen et al. 2004].

Other successful approaches include creating competitions between teams [Beenen et al. 2004] or calling out the number of other people who have also contributed [Salganik and Watts 2009]. Kraut and Resnick's book on online communities provides an extremely thorough reference for this material [Kraut et al. 2012].

3.5. Information seeking and organizational intelligence

Human-computer interaction has long focused on user interaction with many kinds of information. Thus, a second set of key decision points relates to the kind of information we would like to pull from the crowd. For example, in some situations, this information already exists in the heads of other individuals. Mark Ackerman introduced the idea of actively recruiting and gathering this knowledge, through a system called Answer Garden [Ackerman and Malone 1990]. Answer Garden was the precursor to today's question-and-answer (Q&A) systems such as Yahoo! Answers and Quora. It encouraged organization members to create reusable knowledge by asking questions and retaining the answer for the next user who needed it.

Since Answer Garden, these systems have matured into research and products such as the social search engine Aardvark [Horowitz and Kamvar 2010]. Recent work has focused on social search [Evans and Chi 2008; Morris et al. 2010], where users ask their own social networks to help answer a question. This approach of *friendsourcing* [Bernstein et al. 2010b] can solve problems that generic crowds often cannot.

3.6. Exploration and discovery

Crowds bring together diverse perspectives; as Linus Torvalds's saying goes, "given enough eyeballs, all bugs are shallow." Thus, it's not surprising that some of the most influential crowdsourcing communities have focused around discovery.

Mentioned earlier, the FoldIt protein folding game is the most well-known example of crowd discovery. FoldIt is a simulation and puzzle game where players try to fold the a protein's structure as well as possible [FoldIt 2008]. The game has attracted nearly 250,000 players, and their players have uncovered protein folding configurations that have baffled scientists for years [Cooper et al. 2010]. That this result appeared in Nature suggests something about its ability to solve important hard problems.

FoldIt attracted novices. This is not uncommon where the scientific goal holds intrinsic interest: the Galaxy Zoo project, which labels galaxy images from a star survey [Lintott et al. 2008], is another good example. Cooperative crowdsourcing tools may also allow users to go deep and explore micro-areas of interest: for example, collaborative visualization tools such as sense.us [Heer et al. 2007] and ManyEyes [Viegas et al. 2007] allowed users to share visualizations and collaboratively work to explain interesting trends.

3.7. Creativity

Can crowds be creative? Certainly members of that crowd can be. Researchers created the Scratch online community for children to create and remix animations [Resnick et al. 2009]. However, it's not clear that remixing actually produces higher-quality output [Hill and Monroy-Hernández 2013]. In a more mature setting, members of the Newgrounds animation site spend many hours creating collaborative games and animations. These collaborations are delicate and don't always succeed [Luther and Bruckman 2008]. When they do succeed, the onus is on the leader to articulate a clear vision and communicate frequently with participants [Luther et al. 2010].

HCI pursues an understanding of how to best design for the success of creative collaborations. For example, it may be that structuring collaborative roles to reflect the complementary creative strengths of the crowd and the individual can help. Ensemble is a collaborative creative writing application where a leader maintains a high-level vision and articulates creative constraints for the crowd, and the crowd generates text and contributions within those constraints [Kim et al. 2014]. In the domain of music, data from the annual February Album Writing Month (FAWM) uncovered how complementary skill sets can be predictive of successful creative collaborations [Settles and Dow 2013].

3.8. Collective action

Volunteer crowds can come together to affect change in their world. Early in the days of crowdsourcing, this situation hit home with the academic computer science community when a well-known professor at UC Berkeley named Jim Gray disappeared at sea while flying his plane. The community rallied, quickly hacking together software to search satellite images of the region to find Jim's downed plane, where he might still be [Hellerstein and Tennenhouse 2011].

These events were precursor to work that study and collect collective action efforts. However, as with all collective action problems, getting off the ground can be a challenge. Thus, Catalyst allows individuals to condition their participation on others' interest, so that I might commit to tutoring only if ten people commit to attending my tutoring session [Cheng and Bernstein 2014]. We may yet see examples of crowds coming together not just to talk, but to *act*, as Mechanical Turk workers have done on Dynamo (www.wearedynamo.org).

4. PASSIVE CROWDSOURCING

Crowd-sourcing is often perceived as requiring a requester to make direct elicitation of human effort. However, the relationship between requester and the crowd can also be indirect. In *passive crowdsourcing* the crowd produce useful "work product" simply as part of their regular behavior. That is, the work is a *side-effect* of what people were doing ordinarily. Rather than directing the efforts of the crowd as in the *active* scenarios, the requester is *passively* monitoring behavioral traces and leveraging them.

As a simple example, take a Web search engine that collects user logs of search and click behavior (i.e., which results are clicked after the search). The system observes that when most users search for some concept (say, "fruit trees") they conclude their search session on a particular entry in the search result page (say, the 3^{rd} result). From this the system infers that the 3rd page is likely the best answer to the query and the result is boosted to the top of the list [Culliss 2000]. The crowd here is doing work for the "requester"–they are helping organize search results–but this is simply a side-effect of how they would use the system ordinarily, which is to find results of interest.

The difficulty in this approach is that passive crowdsourcing systems are explicitly designed to avoid interfering with the worker's "ordinary" behavior. This requires effective instrumentation, calibration, and inference that allows the system designer to go from a noisy-signal–that is at best weakly connected to the desired work product–to something useful. For example, the search engine does not directly ask the end user to indicate which result is the best, rather they *observe* the click behavior and *infer* the best result.

Passive designs are often used to achieve some effect in the original system (e.g., better search results), but the traces can also be used for completely different applications. For example, companies such as AirSage [AirSage 2014] have utilized the patterns by which cell phones switch from tower to tower as people drive (the "ordinary" behavior that allows cell phones to function) in order to model traffic flow and generate real-time traffic data. In all of these instances there is no explicit "request" being made to the crowd, but the crowd is nonetheless doing work.

4.1. Examples of passive crowd work

The idea of non-reactive measures has a significant history in the sociological literature [Webb et al. 1999] where researchers identified mechanisms for collecting data without directly asking subjects. The quintessential example is the identification of the most popular piece of art in the museum by observing how often different floor tiles needed to be replaced.

The goal of this approach is the capture of specific measures by mining *indirect* measures based on the accretion and erosion behaviors of populations as they move around their daily lives. *Accretion* behaviors are those that involve the mining of created artifacts. This may involve everything from cataloging what people throw in their trash cans and put out on the curb to understand food consumption patterns [Rathje and Murphy 2001] to tracking status updates on Twitter to understand the spread of disease [Sadilek et al. 2012]. The converse, *erosion* patterns, track the (traditionally) physical wear and tear on an object. The replaced floor tiles are an example as is studying the so-called"cow-paths"–physical traces made by populations as they find the best way to get from one place to another (often not the designed, paved solution). Although the notion of "erosion" is less obvious in digital contexts, systems like Waze [Waze 2014] have similarly analyzed individual paths (as measured by cell-phone traces) to identify the fastest route from place to place. The Edit-wear and Read-wear system proposed by [Hill et al. 1992] similarly captured where in a document individuals were spending their time reading or editing.

There are many modern examples for passive crowd-work that leverage social media data. Twitter, Facebook, foursquare, Flickr, and others have all been used as sources of behavioral traces that are utilized for both empirical studies and in the design of systems. A popular application has been the identification of leading indicators for everything from the spread of disease [Sadilek et al. 2012] to political outcomes [Livne et al. 2011]. As individuals signal their health (e.g., "high fever today, staying home") or their political opinions (e.g., "just voted for Obama,") through social media channels, this information can be used to predict the future value of some variable (e.g., number of infections or who will win the election).

Other systems have demonstrated the ability to generate sophisticated labels for physical places by passively observing the traces of individuals. For example, the Livehoods project [Cranshaw et al. 2012] utilizes foursquare checkins to build refined models of geographically based communities, which are often different from the labeled neighborhoods on a map. As individuals wander in their daily lives and report their location to foursquare, the project is able to identify patterns of checkins across a larger population and to identify those new neighborhood structures. Similar projects have utilized geotagged data to identify where tourists go [Flickr 2010] and identify place "semantics" using tagged (both in the textual and geographical sense) images [Rattenbury et al. 2007].

Passive solutions have also leveraged as a means for providing support. For example, the HelpMeOut system [Hartmann et al. 2010] used instrumented Integrated Development Environments (IDEs) as way of logging a developer's reaction to an error. By logging the error and fix, the system could build a database of recommendations that could be provided to future developers encountering the same issue. The Codex system [Fast et al. 2014] identified common programing idioms by analyzing the work-product of developers (millions of lines of Ruby code) to provide labels and warnings to future developers. Query-Feature Graphs [Fourney et al. 2011] mined search logs for common tasks in an end-user application (the image editing program, GIMP). Often people would issue queries such as "how do I remove red-eyes in GIMP." The system found these common queries and by mining the Web identified common commands that were used in response documents. This allowed the system to automatically suggest commands given a high level end-user need. CommandSpace [Adar et al. 2014] extended this idea by jointly modeling both system features and natural language (in this case from Adobe Photoshop). This information was mined from "found" text which included tutorials, message forums, and other traces left by end-users on the Web. By recognizing system features in text and utilizing a distributed vector representation, a number of translations could be supported. For example, finding related features, searching for features based on text, search by analogy, and identifying likely uses of a feature.

4.2. System design

While attractive in that they don't require intervention or disrupting the user, passive crowd work platforms must still be carefully designed. The *inference gap* reflects the fact that many of the observed behaviors are quite distant from the actual work we would like to see performed. That is, we may have an Twitter user saying, "I'm feel terrible today," or a Google searcher looking for "fever medication." However, what the system requester would really like to know is if the person is sick with the Flu today. The further the "instrument" is from what is being measured, the more difficult it is to make the inference. Additionally, many systems and behaviors change over time (e.g., the search engine results change, a social media system is used differently, or the interface adds additional functions or removes others). Consequently, a great deal of care is necessary in such passive systems to ensure that the models and inferences remain predictive over time [Lazer et al. 2014]. Ideally, a passive crowd system would measure behavior in the closest way possible to what is actually the target of measurement, and that any inference be updated.

A second issue to consider is the *reactivity* of the passive solution. That is, when the mined behavioral data is used in a feedback loop inside of the system. For example, a frequently clicked on search-result will move to the top of the search engine result page. However, this will reduce the chance that other, potentially better pages will be identified. Similarly, if the public is aware that tweets are being used to predict elections, their tweeting behavior may change and forecasting accuracy may suffer [Gayo-Avello 2013].

4.3. Ethics

The ethical issues with passive crowd work are somewhat different than their active versions. Those producing work are likely unaware that their traces are being used and for what purposes. The decision of when and how this information is shared is critical. Facebook, for example, ran a field experiment to understand whether seeing positively- or negatively-valenced emotion words in friends' status updates would cause users to use similarly-valenced words in their own posts [Kramer et al. 2014]. They called this a test of "emotional contagion". To test their hypothesis, they randomly hid some status updates matching a list of positive or negative emotion words from the newsfeed, and found a small effect where friends would use more of those same kinds of words if the status update was included in the newsfeed. When Facebook published this result, the mass media and several researchers blasted Facebook for running experiments on emotional contagion without informed consent. The goal of learning about people was overshadowed for these individuals by ethical concerns about online experimentation. The case is an especially fraught one because internet companies such as Google and Facebook run such experiments all the time to improve their products.

In addition, depending on how much is explained, the collection process that was once non-reactive can no longer be perceived as such. The end user being tracked is now aware of the collection and potentially the use of their behavioral traces and may act differently than before. This also opens up the system to creative attacks (e.g., by a search engine optimizer) who may seek to change the way the system operates. Finally, because the worker is unaware that they are doing work they are frequently unpaid (at least through direct compensation). These considerations must be weighed when passive crowd work is used.

5. CHALLENGES IN CROWDSOURCING

Human-computer interaction is helping to shape the future of crowdsourcing through its design of the technology that people will use to engage with crowdsoucing as either requesters and crowd workers. Over the past few years, the field has become aware that the problems that it choose to focus on or not may have a very real impact on not only the benefits we stand to gain through crowdsourcing but also the impact that it may have on how people choose to work in the future. Kittur et al. engage in a thorough exploration of these challenges [Kittur et al. 2013]. It asks: What would it take for us to be comfortable with our own children becoming fulltime crowd workers?

Since the earliest days of human computation, its proponents have discussed how the eventual goal is to develop hybrid systems that engage with both humans intelligence drawn from the crowd and machine intelligence realized through artificial intelligence and machine learning. This vision remains, but systems still utilize it in very basic ways. One of our visions for crowdsourcing in the future is one in which truly intelligent systems are developed more quickly by initially create crowd-powered systems and then using them as scaffolding to gradually move to fully automated approaches.

Crowdsourcing has traditionally worked best, although not exclusively, for problems that required little expertise. A challenge going forward is to push on the scope of problems possible to solve with crowdsourcing by engaging with expert crowds, embedding needed expertise in the tools non-expert crowds use, or by using a flexible combination of the two (e.g., [Retelny et al. 2014]).

As more people participate as crowd workers, it is becoming increasingly important to understand this component of the labor force and what tools might be useful to create to help not only requesters but also workers. Workers on many crowd marketplaces face inefficiencies that could be improved with better tools, such as finding tasks that are best suited to their skills. It is also difficult for workers today to be rewarded over time for acquiring expertise in a particular kind of crowd work.

6. CONCLUSION

Human-computer interaction has contributed to crowdsourcing by creating tools allowing different stakeholders to participate more easily or more powerfully, and understanding how people participate in order to shape a brighter future for crowd work. One of the reasons that crowdsourcing is interesting is because technology is allowing groups to work together in ways that were infeasible only a few years ago. The challenges going forward are to ensure that requesters and workers are able to realize the potential of crowdsourcing without succumbing to its potential downsides, and to continue to improve the systems enabling all of this so that even more is possible.

Acknowledgments

The authors would like to thank Rob Miller for his early brainstorms and framing organization for this chapter.

REFERENCES

Mark S Ackerman and Thomas W Malone. 1990. Answer Garden: a tool for growing organizational memory. In *Proc. GROUP '90*. http://portal.acm.org/citation.cfm?id=91474.91485

Eytan Adar, Mira Dontcheva, and Gierad Laput. 2014. CommandSpace: Modeling the Relationships Between Tasks, Descriptions and Features. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 167–176. DOI:http://dx.doi.org/10.1145/2642918.2647395

Salman Ahmad, Alexis Battle, Zahan Malkani, and Sepander Kamvar. 2011. The jabberwocky programming environment for structured social computing. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 53–64.

AirSage. 2014. AirSage. http://www.airsage.com. (2014). Accessed: 2014-04-22.

Judd Antin and Aaron Shaw. 2012. Social desirability bias and self-reports of motivation: a cross-cultural study of Amazon Mechanical Turk in the US and India. In *Proc. CHI '12*. DOI:http://dx.doi.org/10.1145/2208636.2208699

Gerard Beenen, Kimberly Ling, Xiaoqing Wang, Klarissa Chang, Dan Frankowski, Paul Resnick, and Robert E. Kraut. 2004. Using Social Psychology to Motivate Contributions to Online Communities. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04)*. ACM, New York, NY, USA, 212–221. DOI:http://dx.doi.org/10.1145/1031607.1031642

Michael S Bernstein, Joel Brandt, Robert C Miller, and David R Karger. 2011. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 33–42.

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

Michael S Bernstein, Desney Tan, Greg Smith, Mary Czerwinski, and Eric Horvitz. 2010b. Personalization via friendsourcing. *ACM Transactions on Computer-Human Interaction (TOCHI)* 17, 2 (2010), 6.

Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatrowicz, Brandyn White, Samuel White, and Tom Yeh. 2010. VizWiz: nearly real-time answers to visual questions. In *Proc. UIST '10*.

Noah Burbank, Debojyoti Dutta, Ashish Goel, David Lee, Eli Marschner, and Narayanan Shivakumar. 2011. Widescope-A social platform for serious conversations on the Web. *arXiv preprint arXiv:1111.1958* (2011).

Justin Cheng and Michael Bernstein. 2014. Catalyst: Triggering Collective Action with Thresholds. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, New York, NY, USA, 1211–1221. DOI:http://dx.doi.org/10.1145/2531602.2531635

Ed H Chi. 2009. A position paper onliving laboratories: Rethinking ecological designs and experimentation in human-computer interaction. In *Human-Computer Interaction. New Trends*. Springer, 597–605.

Lydia B Chilton, Greg Little, Darren Edge, Daniel S Weld, and James A Landay. 2013. Cascade: Crowdsourcing taxonomy creation. In *Proceedings of the 2013 ACM annual conference on Human factors in computing systems*. ACM, 1999–2008.

Lydia B Chilton, Clayton T Sims, Max Goldman, Greg Little, and Robert C Miller. 2009. Seaweed: A web application for designing economic games. In *Proceedings of the ACM SIGKDD workshop on human computation*. ACM, 34–35.

Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and Foldit-Players. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466, 7307 (2010), 756–760. http://www.nature.com/nature/journal/v466/n7307/pdf/nature09304.pdf?utm_source=twitterfeed& utm_medium=twitter

Justin Cranshaw and Aniket Kittur. 2011a. The polymath project: lessons from a successful online collaboration in mathematics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1865–1874.

Justin Cranshaw and Aniket Kittur. 2011b. The polymath project: lessons from a successful online collaboration in mathematics. In *Proc. CHI '11*. ACM.

Justin Cranshaw, Raz Schwartz, Jason I Hong, and Norman M Sadeh. 2012. The Livehoods Project: Utilizing Social Media to Understand the Dynamics of a City.. In *ICWSM*.

G. Culliss. 2000. Method for organizing information. (June 20 2000). http://www.google.com/patents/US6078916 US Patent 6,078,916.

Brynn M Evans and Ed H Chi. 2008. Towards a model of understanding social search. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*. ACM, 485–494.

Siamak Faridani, Ephrat Bitton, Kimiko Ryokai, and Ken Goldberg. 2010. Opinion space: a scalable tool for browsing online comments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1175–1184.

Ethan Fast, Daniel Steffee, Lucy Wang, Joel Brandt, and Michael S Bernstein. 2014. Emergent, Crowd-scale Programming Practice in the IDE. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1–10.

Flickr. 2010. Locals and Tourists - a set on Flickr. https://www.flickr.com/photos/walkingsf/sets/72157624209158632/. (2010). Accessed: 2014-04-22.

FoldIt. 2008. FoldIt: Solve Puzzles for Science. http://fold.it. (May 2008). Accessed: 2014-04-22.

Adam Fourney, Richard Mann, and Michael Terry. 2011. Query-feature graphs: bridging user vocabulary and system functionality. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 207–216.

Daniel Gayo-Avello. 2013. A meta-analysis of state-of-the-art electoral prediction from Twitter data. *Social Science Computer Review* 31, 6 (2013), 649–679.

Elizabeth M Gerber, Julie S Hui, and Pei-Yi Kuo. 2012. Crowdfunding: why people are motivated to post and fund projects on crowdfunding platforms. In *Proceedings of the International Workshop on Design, Influence, and Social Technologies: Techniques, Impacts and Ethics*.

Eric Gilbert. 2013. Widespread underprovision on reddit. In *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 803–808.

Severin Hacker and Luis von Ahn. 2012. *Duolingo*. www.duolingo.com. www.duolingo.com

Björn Hartmann, Daniel MacDougall, Joel Brandt, and Scott R Klemmer. 2010. What would other programmers do: suggesting solutions to error messages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1019–1028.

Jeffrey Heer, Fernanda B Viégas, and Martin Wattenberg. 2007. Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1029–1038.

Joseph M Hellerstein and David L Tennenhouse. 2011. Searching for Jim Gray: a technical overview. *Communcations of the ACM* 54, 7 (July 2011), 77–87. DOI:http://dx.doi.org/10.1145/1965724.1965744

Benjamin Mako Hill and Andrés Monroy-Hernández. 2013. The cost of collaboration for code and art: Evidence from a remixing community. In *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 1035–1046.

William C Hill, James D Hollan, Dave Wroblewski, and Tim McCandless. 1992. Edit wear and read wear. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 3–9.

Damon Horowitz and Sepandar D Kamvar. 2010. The anatomy of a large-scale social search engine. In *Proceedings of the 19th international conference on World wide web*. ACM, 431–440.

Lilly C Irani and M Silberman. 2013. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 611–620.

B Kanefsky, N.G. Barlow, and V.C. Gulick. 2001. Can Distributed Volunteers Accomplish Massive Data Analysis Tasks?. In *Lunar and Planetary Institute Science Conference Abstracts (Lunar and Planetary Inst. Technical Report)*, Vol. 32.

Brian Keegan and Darren Gergle. 2010. Egalitarians at the gate: One-sided gatekeeping practices in social media. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. ACM, 131–134.

Firas Khatib, Frank DiMaio, Seth Cooper, Maciej Kazmierczyk, Miroslaw Gilski, Szymon Krzywda, Helena Zabranska, Iva Pichova, James Thompson, Zoran Popović, and others. 2011. Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature structural & molecular biology* 18, 10 (2011), 1175–1177.

Joy Kim, Justin Cheng, and Michael S. Bernstein. 2014. Ensemble: Exploring Complementary Strengths of Leaders and Crowds in Creative Collaboration. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, New York, NY, USA, 745–755. DOI:http://dx.doi.org/10.1145/2531602.2531638

Aniket Kittur and Robert E Kraut. 2008. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*. ACM, 37–46.

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

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

Aniket Kittur, Bongwon Suh, Bryan A. Pendleton, and Ed H. Chi. 2007. He says, she says: conflict and coordination in Wikipedia. In *Proc. CHI '07*. http://portal.acm.org/citation.cfm?id=1240624.1240698

Adam DI Kramer, Jamie E Guillory, and Jeffrey T Hancock. 2014. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences* (2014), 201320040.

Robert E Kraut, Paul Resnick, Sara Kiesler, Moira Burke, Yan Chen, Niki Kittur, Joseph Konstan, Yuqing Ren, and John Riedl. 2012. *Building successful online communities: Evidence-based social design*. MIT Press.

Travis Kriplean, Jonathan T Morgan, Deen Freelon, Alan Borning, and Lance Bennett. 2011a. ConsiderIt: Improving structured public deliberation. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1831–1836.

T Kriplean, M Toomim, JT Morgan, A Borning, and AJ Ko. 2011b. REFLECT: Supporting active listening and grounding on the Web through restatement. *Proc. CSCW11 Horizon* (2011).

Anand P Kulkarni, Matthew Can, and Bjoern Hartmann. 2011. Turkomatic: automatic recursive task and workflow design for mechanical turk. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2053–2058.

Walter Lasecki, Christopher Miller, Adam Sadilek, Andrew Abumoussa, Donato Borrello, Raja Kushalnagar, and Jeffrey Bigham. 2012. Real-time captioning by groups of non-experts. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 23–34.

Walter S Lasecki, Christopher D Miller, and Jeffrey P Bigham. 2013. Warping time for more effective real-time crowdsourcing. In *Proceedings of the 2013 ACM annual conference on Human factors in computing systems*. ACM, 2033–2036.

Walter S Lasecki, Kyle I Murray, Samuel White, Robert C Miller, and Jeffrey P Bigham. 2011. Real-time crowd control of existing interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 23–32.

Jean Lave and Etienne Wenger. 1991. *Situated learning: Legitimate peripheral participation*. Cambridge university press.

David M Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. 2014. The Parable of Google Flu: Traps in Big Data Analysis. (2014).

Matthew Lease, Jessica Hullman, Jeffrey P Bigham, Michael Bernstein, J Kim, W Lasecki, S Bakhshi, T Mitra, and RC Miller. 2013. Mechanical turk is not anonymous. *Social Science Research Network* (2013).

Chris J Lintott, Kevin Schawinski, Anže Slosar, Kate Land, Steven Bamford, Daniel Thomas, M Jordan Raddick, Robert C Nichol, Alex Szalay, Dan Andreescu, and others. 2008. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society* 389, 3 (2008), 1179–1189.

Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. 2009. Turkit: tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*. ACM, 29–30.

Avishay Livne, Matthew P Simmons, Eytan Adar, and Lada A Adamic. 2011. The Party Is Over Here: Structure and Content in the 2010 Election.. In *ICWSM*.

Richard Louv, John W Fitzpatrick, Janis L Dickinson, and Rick Bonney. 2012. *Citizen science: Public participation in environmental research*. Cornell University Press.

Kurt Luther and Amy Bruckman. 2008. Leadership in online creative collaboration. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*. ACM, 343–352.

Kurt Luther, Kelly Caine, Kevin Ziegler, and Amy Bruckman. 2010. Why it works (when it works): success factors in online creative collaboration. In *Proceedings of the 16th ACM international conference on Supporting group work*. ACM, 1–10.

Kurt Luther, Casey Fiesler, and Amy Bruckman. 2013. Redistributing leadership in online creative collaboration. In *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 1007–1022.

Winter Mason and Duncan J Watts. 2010. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter* 11, 2 (2010), 100–108.

Jonathan T. Morgan, Siko Bouterse, Heather Walls, and Sarah Stierch. 2013. Tea and Sympathy: Crafting Positive New User Experiences on Wikipedia. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. ACM, New York, NY, USA, 839–848. DOI:http://dx.doi.org/10.1145/2441776.2441871

Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. 2010. What do people ask their social networks, and why?: a survey study of status message q&a behavior. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1739–1748.

Sean A Munson, Stephanie Y Lee, and Paul Resnick. 2013. Encouraging reading of diverse political viewpoints with a browser widget. *Proc. ICWSM 2013* (2013).

Iftekhar Naim, Daniel Gildea, Walter Lasecki, and Jeffrey P Bigham. 2013. Text alignment for real-time crowd captioning. In *Proceedings of NAACL-HLT*. 201–210.

E V Nalimov, C Wirth, G M C Haworth, and Others. 1999. KQQKQQ and the Kasparov-World Game. *ICGA Journal* 22, 4 (1999), 195–212.

Jon Noronha, Eric Hysen, Haoqi Zhang, and Krzysztof Z Gajos. 2011. Platemate: crowdsourcing nutritional analysis from food photographs. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 1–12.

Ory Okolloh. 2009. Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information. *Participatory Learning and Action* 59, 1 (2009), 65–70.

Katherine Panciera, Aaron Halfaker, and Loren Terveen. 2009. Wikipedians are born, not made: a study of power editors on Wikipedia. In *Proceedings of the ACM 2009 international conference on Supporting group work*. ACM, 51–60.

Jennifer Preece and Ben Shneiderman. 2009. The reader-to-leader framework: Motivating technology-mediated social participation. *AIS Transactions on Human-Computer Interaction* 1, 1 (2009), 13–32.

Alexander J Quinn and Benjamin B Bederson. 2014. AskSheet: Efficient Human Computation for Decision Making with Spreadsheets. In *Proc. of ACM Conf. on Computer Supported Cooperative Work (CSCW 14)*.

William L Rathje and Cullen Murphy. 2001. *Rubbish!: the archaeology of garbage*. University of Arizona Press.

Tye Rattenbury, Nathaniel Good, and Mor Naaman. 2007. Towards automatic extraction of event and place semantics from flickr tags. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 103–110.

Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and others. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.

Daniela Retelny, Sebastien Robaszkiewicz, Alexandra To, Walter Lasecki, Jay Patel, Negar Rahmati, Tulsee Doshi, Melissa Valentine, and Michael S. Bernstein. 2014. Expert Crowdsourcing with Flash Teams. In *Proc. UIST '14*.

Brian C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. 2008. LabelMe: a database and web-based tool for image annotation. *International journal of computer vision* 77, 1 (2008), 157–173.

Richard M Ryan and Edward L Deci. 2000. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist* 55, 1 (2000), 68.

Jeffrey Rzeszotarski and Aniket Kittur. 2012. CrowdScape: interactively visualizing user behavior and output. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 55–62.

Adam Sadilek, Henry A Kautz, and Vincent Silenzio. 2012. Modeling Spread of Disease from Social Interactions.. In *ICWSM*.

Matthew J Salganik and Duncan J Watts. 2009. Web-Based Experiments for the Study of Collective Social Dynamics in Cultural Markets. *Topics in Cognitive Science* 1, 3 (2009), 439–468.

Burr Settles and Steven Dow. 2013. Let's get together: the formation and success of online creative collaborations. In *Proceedings of the 2013 ACM annual conference on Human factors in computing systems*. ACM, 2009–2018.

Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open Mind Common Sense: Knowledge acquisition from the general public. *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE* (2002), 1223–1237.

Kate Starbird. 2013. Delivering patients to sacré coeur: collective intelligence in digital volunteer communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 801–810.

Kathleen Tuite, Noah Snavely, Dun-yu Hsiao, Nadine Tabing, and Zoran Popović. 2011. PhotoCity: Training experts at large-scale image acquisition through a competitive game. In *Proc. CHI '11*. ACM.

Fernanda B Viegas, Martin Wattenberg, Frank Van Ham, Jesse Kriss, and Matt McKeon. 2007. Manyeyes: a site for visualization at internet scale. *Visualization and Computer Graphics, IEEE Transactions on* 13, 6 (2007), 1121–1128.

Sarah Vieweg, Amanda L Hughes, Kate Starbird, and Leysia Palen. 2010. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1079–1088.

Luis Von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 319–326.

Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. 2008. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science* 321, 5895 (2008), 1465–1468. DOI:http://dx.doi.org/10.1126/science.1160379

Waze. 2014. Waze. http://waze.com. (2014). Accessed: 2014-04-22.

E.J. Webb, D.T. Campbell, R.D. Schwartz, and L. Sechrest. 1999. *Unobtrusive Measures*. SAGE Publications. http://books.google.pl/books?id=z2u8QgAACAAJ

Draft as of 11th January, 2015