Hired Hands and Dubious Guesses:
Adventures in Crowdsourced Data Collection

Aaron Shaw

Introduction

"What if we used Mechanical Turk to do it?" It was one of those hare-brained, off-handed suggestions that bubbled up frequently enough in our brainstorming sessions that I no longer remember exactly who said it. There were three of us around the conference table that day, our diagrams and notes on the floor-to-ceiling dry-erase boards, our laptops, papers, and cellphones spread across the table. As soon as the sentence came out, smiles spread across each of our faces. The faculty principal investigator on the project started to chuckle out loud. The laughter spread and in a few seconds we were mapping out the hypothetical steps and implications of the idea. We left the room still talking, the conversation light with excitement and anticipation.

Something about the idea of crowdsourcing our data collection seemed completely appropriate. We had struggled for weeks to arrive at a compelling research design and found ourselves confronting a very concrete data collection

problem: how to do an extensive content analysis of several hundred websites at once. The questions we wanted to ask about each site were fairly simple, but far too numerous to make it practical to answer them all without the help of a very large team of research assistants. Knowing that we didn't really have the organizational or financial resources to undertake that effort, we had started to consider scaling back the study and other alternatives. Then came the Mechanical Turk idea: we thought we would distribute the work as a massive set of micro-tasks through an online labor market. As it turned out, we had no idea what that would entail.

Mechanical Turk is a Web service operated by Amazon.com and the most widely known commercial platform for distributed work online, which is also known as crowdsourcing or human computation. A decent definition of crowdsourcing is: *the disaggregation and distribution of large tasks across a large group of people over the Internet*.[1] To illustrate what this actually means, imagine that you have 100 million photos that you need to make sure do not include any

---

[1] As with many neologisms related to the Internet, competing definitions for crowdsourcing exist. Mine is derived from a combination of the original description by Howe (2008), who coined the term, and my own observations.

child pornography, hate speech, copyright violations, or something similar.[2] The automated methods of filtering images for this kind of stuff turn out to be imperfect in a bunch of ways, so you cannot trust them 100% of the time. As a result, you need to verify that these automated methods work by at least reviewing a sample of the images by hand. Crowdsourcing the solution to this problem would entail that you use some system (probably also online) to enable as many people as possible to review as few or as many photos at a time as they want (probably in exchange for very tiny amounts of money for every photo reviewed).[3] Other examples of

---

[2] As it happens, this is roughly the number of photos uploaded to Facebook *every single day*. Source: http://en-gb.facebook.com/notes/facebook-engineering/developing-facebooks-new-photo-viewer/499447633919. To give you some tangible idea of how many photos this is, consider that 100 million sheets of regular thickness (0.0004 in. or 0.01 cm) paper would create a stack roughly 30,000 feet (or 10 km) high.

[3] This kind of arrangement raises all kinds of questions, like how you know that the photo reviewers are not also the kind of people who like to post porn everywhere they go on the Internet; how you figure out what to do when two reviewers of the same photo disagree about what constitutes porn in the first place; and whether or not its ethically a good thing for anybody to be doing mind-numbing work like this

crowdsourcing and human computation involve very different kinds of problems and people, as well as more or less complicated procedures.

Over the past five years or so, crowdsourcing of many different kinds has become a growing area of investment and innovation in Silicon Valley as well as in several research communities. At the same time, crowdsourcing remains a relatively foreign concept among social scientists. The purpose of this chapter is to both introduce the idea of using crowdsourcing for social science research and to address some of the challenges I have encountered in trying to actually do so. Despite the fact that writing something like this makes it seem like I am setting myself up as an expert on the subject, I have really been more of a spectator sitting somewhere off to the side of the *avant garde* of the crowdsourcing industry and the research communities involved in advancing the development of new techniques and tools for managing "collective intelligence." In the process, I have developed multiple research projects that utilize crowdsourcing both as a field site where I have recruited research subjects as well as a source for distributed research

---

in exchange for such small amounts of money; etc. These are hard questions that are fundamental to crowdsourcing as both a social technology and an economic phenomenon. I'll try to address some of them below.

assistance. In the material that follows, I discuss practical issues involved in this work that extend beyond such a novel, unusual field of study.

The biggest concerns with crowdsourcing any aspect of a research project boil down to the fact that the practice of crowdsourcing tends to be (1) organized like poorly-compensated online piece-work, and (2) very boring. In addition, customizing many of the interfaces through which online crowdsourcing takes place can require significant computer programming knowledge, and some of us who choose to use them enlist the help of a software engineer at some point. As a result, you, the researcher, wind up confronting digital-age variants of organizational problems that have plagued large-scale divisions of scientific, industrial, and bureaucratic labor for several centuries. How do you hire and manage employees to contribute to a project without sacrificing quality? How do you collaborate with colleagues who have completely different training and vocabulary from your own? How do you learn to use new tools without wasting resources? None of my graduate theory and methods seminars prepared me adequately to confront these challenges, despite the fact that I now believe they have been commonplace in academic research for a long time.

In a classic 1966 essay, Julius Roth described the problems with using what he called "hired hands" to conduct research work.[4] His analysis speaks to a general problem of any research project employing multiple people and technologies to perform data collection or analysis: "When the tasks of a research project are split up into small pieces to be assigned to hired hands, none of these data-collectors and processors will ever understand all the complexities and subtleties of the research issues in the same way as the person who conceived of the study...Since the [research] director often cannot be sure what conceptions of the issues the hired hands have as a result of his explanations and "training," he must make *dubious guesses* about the meaning of much of the data they return to him...As he gains in quantity of data, he loses in validity and meaningfulness." (Roth 1966: 193, *emphasis added*).

Roth could just as easily be speaking of a more typical study with one or two paid research assistants as a crowdsourcing study employing thousands on Mechanical Turk or some other web-based platform. Irrespective of the type or scale of work you do, the critique (as well as his humorous ethnographic examples of hired hands undermining researchers' objectives) should strike terror into your

---

[4]Thanks to the editors of this volume for pointing me to Roth's essay.

heart if you had assumed that such farming out your scholarship to people or machines was going to be a straightforward affair.

At the same time, I disagree with part of Roth's premise as well as his conclusion. Problems of data quality and uncertainty apply even when research assistants do not perform your data collection for you. The difficulties of calibrating and coordinating diverse sets of people, socio-technical systems, and techniques affect even those projects where there is only one researcher involved. Data collection methods and artifacts may or may not, in Langdon Winner's (1986) famous phrase, "have politics" but they are also far from neutral instruments that exist only to serve the scholarly will in a direct or unmediated way. It takes a lot of practical learning and work to apply any method effectively, and the process usually entails numerous failures and mistakes. Furthermore, the quality of data does not necessarily increase in inverse proportion to its quantity. The failures, mistakes, and "dubious guesses" that so often look like bad research work often feed into the construction of more nuanced, intelligent understanding of a particular process, method, or person.

A narrower form of my argument may be more straightforward: *creating and managing an effective division of intellectual labor lies at the heart of any ambitious research project*. By "research project" and "intellectual labor" here I really mean the whole gamut – all the stuff that researchers, research assistants,

colleagues and supervisors do in addition to the methods and the tools involved in data collection and analysis. For a variety of reasons I explain below, crowdsourcing has illustrated the value of this broad perspective to me many times over. As social scientists, many of us may cling to the myth of the heroic scholar, laboring away in monastic, dusty solitude and publishing brilliant (sole-authored!) articles or books. However, the reality of scholarly labor is almost always more interesting and complicated, involving teams of assistants, administrative staff, librarians, statisticians or – increasingly for those of us doing work involving digital media – software, engineers, web sites, and Internet users about whom we may know shockingly little. The names or accomplishments of these individuals or tools may appear in footnotes and acknowledgements, but the labor of their contributions often remains invisible, with the heavy brush strokes of collaboration, paid work, and research administration obscured behind a polite veneer of attribution.

Scrutinizing the division of academic labor from this point of view can get a little uncomfortable. What began in my research group meeting that day as a starry-eyed foray into crowdsourcing research has forced me to recognize some of my shortcomings as a scholar as well as the extent to which I have capitalized on the contributions of others to my work in a way that feels unfair and even a little exploitative. At the same time, I still hope that these realizations may make me a better researcher in the end. Crowdsourcing has shown me that in the course of a

collective endeavor, almost everyone and everything involved will, at some point, make "contributions" that are either unproductive or an active hindrance to the progress of the project. The interesting questions arise when you try to figure out how to learn from all these failures and "dubious guesses."

Crowdsourcing and the Division of Scientific Labor

As I suggested above, the need to divide an overwhelming amount of academic labor drove my collaborators and I to use crowdsourcing for our data collection. In this regard, our motives resemble the reasons human computation and crowdsourcing were developed in the first place. Historical examples of the division of academic labor overlap with the earliest cases of human computation, which in turn gave rise to contemporary crowdsourcing techniques.[5] In 1759, a team of three French astronomers used calculus to work out an accurate estimate for the date on which Halley's Comet would pass closest to Sun. In doing so, the three, Alexis Clairaut, Joseph Lalande, and Nicole-Reine Lepaute, had to perform literally thousands of arithmetic calculations *by hand*, a task that they completed in part by dividing the labor and devising systematic procedures to check for errors. The

---

[5]My account here leans heavily on David Alan Grier's *When Computers Were Human* (2005).

techniques they developed became the foundation for subsequent practices of scientific calculation and computation, which grew in scope and ambition as calculus and statistics became prominent fields of research and innovation. Following the two World Wars, during which human computers even calculated the ballistics tables used in the field to aim various weapons, human computation waned in prominence as mainframe computing became feasible and outpaced the processing power of people alone.[6]

---

[6]It is worth noting that from the late 19th century until the advent of the mainframe computing era, the majority of *human* computers were women (Grier 2005). This kind of academic grunt work provided one of the few avenues of scientific achievement accessible to (and culturally acceptable for) women at the time. In all likelihood, the relative invisibility of human computers in the history of science and mathematics derives from this *feminization* of computational labor. In addition, the similarly forgotten fact that the earliest computer programmers were women resulted directly from the legacy of women in human computation. Just as the equipment and research budgets for computing got bigger, the number of women involved in the field declined.

Ironically, the diffusion of digital computing networks that followed the mainframe era has also brought with it a renaissance of human computing, more commonly referred to now as crowdsourcing. As was the case with human computation, much of the innovation in crowdsourcing has happened through scientific research. The 2001 NASA clickworkers project, in which visitors to a NASA website could volunteer to label craters in photos of the surface of Mars, represents the foundational example of contemporary academic crowdsourcing (Benkler, 2006: 136-138; Kanefsky, Barlow and Gulick, 2001).[7] Since then, a number of scientific projects have adopted similar techniques to engage amateur citizen scientists in data collection and classification.[8] In addition, a growing

---

[7] In that case, three NASA researchers, Bob Kanefsky, Nadine Barlow, and Virginia Gulick, wanted to identify craters in photographs of the surface of Mars taken by the Viking orbiter and decided to see whether volunteers recruited over the Internet could perform the task as well as a physicist with a Ph.D (Barlow herself). The results of their study (which they conceived as a side project) suggested that with a little bit of filtering, the accuracy of the results provided by the volunteers was about the same as that of the physicist (Kanefsky, Barlow, and Gulick, 2001).

[8] I should point out that some of the creators of these projects do not call what they do crowdsourcing and argue that there are fundamental differences between

number of commercial crowdsourcing platforms have made paid, distributed work more accessible. Using these platforms, computer scientists and computational linguists have begun to apply more refined filtering techniques to some of the data collected by crowds in order to improve the precision of results (e.g., Quinn and Bederson, 2011; Sheng, Provost, and Ipeirotis, 2008, Snow, O'Connor, Jurafsky, and Ng, 2008). The result has been a concomitant growth in research on the dynamics and applications of crowdsourcing, including a growing body of work that uses crowdsourcing for social scientific inquiry and to address topics of relevance to social scientists (Horton, Rand, and Zeckhauser, 2010).

There are now several websites that make it feasible to recruit a crowd of your own (in exchange for payment or not) and indeed, several companies have received venture funding for business models based on commercial crowdsourcing of various kinds. In terms of research, the diffusion of these tools and techniques has made it so that you no longer need to be an engineer or a physicist at NASA to experiment with crowdsourcing. Indeed, a few dollars and an hour or so of tinkering

---

crowdsourcing and citizen science. I think this is fine, but since I prefer a broader definition of crowdsourcing (see above), I tend not to observe such fine-grained distinctions.

can get you access to a crowd of your own through web services like Amazon's Mechanical Turk, CrowdFlower, or Clickworker (a private company founded in Germany that shares the name of the original NASA project).[9] Likewise, the widespread availability of tools for building web applications make it so that a few hours of programming from a skilled developer are all that stand between you and your own custom-built crowdsourcing platform...in theory.

User Error

Thinking back to the conversation in the conference room, I can see that my colleagues and I had bought into a number of prevailing theories about crowdsourcing, most of which were wrong. To start with, consider the basic insight behind using crowds for any kind of work: that it can be faster and more effective to recruit a large group of non-specialist people to perform a research task in a distributed fashion over the Internet than to employ either a small number of highly-trained research assistants or computer algorithms. I do not consider this statement untrue, but it leaves out so many important details as to be totally useless

---

[9] Full disclosure: I have done paid consulting work with CrowdFlower, which is based in San Francisco, CA.

when you actually try to crowdsource something. I learned just how useless as soon as I began trying to execute our research ideas on Mechanical Turk.

Once we had secured human subjects approval for our project, I started running pilot jobs on Mechanical Turk. Immediately, I found numerous ways to mess them up. My problems started with simple interface design mistakes, such as asking the workers to enter a date in response to a question, but not requiring that their answer be in a particular format. The resulting data was a mess of words, numbers and variations of month-day-year formats I had never seen. It also took me a while to internalize some of the terminology Amazon uses to distinguish between the individual micro-tasks that make up a larger job (they call these “Human Intelligence Tasks″ or “HITs″ and the larger job itself (these used to be known as tasks,but are now called groups. As a result, I ordered several jobs at once in which each of ten questions were answered one hundred times by a single Mturk worker, instead of one hundred workers answering ten questions one time each (feel free to read that over if you need to, or just trust me when I say it's a totally bone-headed move). Luckily, within a few minutes, one of the workers emailed through the Mturk interface to alert me to the situation and I stopped the job after only two hundred or so of the HITs had been completed. Such mistakes were (and still are!) embarrassing as well as costly. I had not accounted for my own ignorance in the original research budget, and after accumulating a handful of similar screw ups, I

had to go to the project Principal Investigator and explain all the idiotic ways in which I had managed to waste our money in order to ask for some additional funds. Even though Mturk tasks cost about a penny each, when you purchase several dozen or hundreds of them at a time and the workers respond to your tasks within seconds, the pennies add up fast.

All of these failures forced me to confront the fact that even a crowd will not do your work for you automatically. Crowdsourcing is only as fast and effective as the person designing and managing the task. The idea that you can conduct large scale projects more efficiently simply by hiring more people falls apart if you do not know how to take advantage of their attention and intelligence. Indeed, my experience suggests a social science corollary to Brook's Law[10]: adding manpower to large research project likely makes it slower.

Communication Breakdowns

Eventually, after completing a few successful pilots, I ran a pair of controlled experiments on Mechanical Turk. For both of these studies, I worked in

---

[10]Brooks (1995) law states: "Adding manpower to a late software project only makes it later."

collaboration with multiple researchers across different disciplines and organizations, using Mechanical Turk as a subject pool to recruit participants (Shaw, Horton, and Chen, 2011, Antin and Shaw, 2011). At the same time, my coauthors and I conducted the experiments using different platforms. The first involved a web-based application that software developers and researchers in a university setting had designed to handle traditional survey and content analysis tasks. The second study ran on the CrowdFlower platform, originally built for the purpose of commercial, enterprise-scale crowdsourcing jobs. In both cases, there were several unanticipated hangups in the process of actually getting to the point where we could run the study. Some of these problems stemmed from using two relatively new pieces of software that still had numerous bugs.[11] Others emerged later, the products of communication breakdowns between myself and the software engineers.

In both experiments, the biggest communication breakdowns concerned the process of assigning participants into groups. It is only a slight exaggeration to say

---

[11] A very distinguished computer science professor only recently explained to me that even the greatest software only has *fewer* bugs; it is never bug free. I didn't fully understand this until I started collaborating with developers in the process of designing, testing and improving real code.

that *random* and *unique* treatment assignment represent the most fundamental prerequisites for a valid controlled experiment in the modern sense of the term (see Fisher, 1935). Basically, this means that each subject in the study gets randomly assigned and exposed to one, and only one, of the experimental conditions. Meeting these criteria help ensure that the study does not violate the *Stable Unit Treatment Value Assumption* (a.k.a. "SUTVA").[12] The bottom line here is that if you mess up either the random or the unique assignment of treatment and control conditions, your experiment cannot generate valid inferences about either any of the differences between the effects of the experimental conditions or any effects the treatment might or might not have on the population at large. In other words, you cannot claim that your experimental treatment caused the results that you found. You can only claim to understand what the results of a controlled experiment mean if you handle the treatment assignment in a systematic way. No exceptions.

Not surprisingly, my collaborators and I wanted to ensure that our engineer collaborators really understood what we meant when we asked them whether the

---

[12] Since I find the term completely obfuscatory and unhelpful, I avoid using it, but I feel sort of obligated to mention it in case you want to look into this kind of thing further or intimidate your friends with academic jargon. The assumptions underlying SUTVA get pretty abstruse and discussing them would take me deep into the methodological weeds, so I'm going to just refer you to some really clear articles that explain how it works in more detail (Holland, 1986; Little & Rubin, 2000).

software really met the requirements of random and unique treatment assignment. For the most part, we, the social scientists, did not really try to understand exactly how the web applications did what they did; however, given the stakes of getting treatment assignment right, we felt we needed to be absolutely certain we understood the steps involved. We would not drop the issue until we were 100% certain that everything worked exactly as we expected 100% of the time.

Needless to say, our software engineer colleagues on both projects found this sort of nit-picky insistence tiresome. On the first experiment with the software developed in an academic research center, hashing out the details of the randomization function generated epic email threads, spanning several weeks and incorporating dozens of examples as all of us tried repeatedly to illustrate our ideas in jargon-free terms. It also resulted in a few more failed pilot studies as we tested and re-tested the software to make sure it behaved as planned. One afternoon, when we were painfully close to running the study, the lead developer and I spent three hours on the phone, walking through every step of the experiment (as I wanted it to run) and every block of the treatment assignment code (as he had written it). This sort of uncompromising attention to detail allowed the study to succeed eventually, but I still came away wondering why we had taken so incredibly long to do so.

On the surface, my experience working with the private startup *looked* totally different. In that case, I had reached an informal agreement with one of the

co-founders to help the company implement some survey work on behalf a prospective client. At the time, the company was extremely small (about five employees) and one of the perks of working with them was that they were excited to help me run experiments using their platform.[13] The three members of the engineering team all had experience building commercial web applications and the platform already had an extremely robust set of features as well as an elegant user interface. Everything suggested that the process would be much smoother and faster than my experience with the small academic research center's software developers.

Despite the platform's powerful ability to scale and all the amazing features that were already built in, it still took almost three months of back and forth with one particularly friendly engineer for me to actually run a controlled experiment. Once again, I spent a good part of this time struggling to explain exactly what I meant by the concept of treatment assignment. Instead of randomization, though, the hangup was around ensuring that each worker saw *one and only one* of the treatments. As with the my first Mechanical Turk pilots, fuzzy distinctions between

---

[13] In this regard, my experience collaborating on research with a private sector organization had some interesting parallels with the discussion by Williams and Xiong (2009). Like them, I found it necessary to consciously frame my work to be more interesting to an audience oriented towards very practical problems of running a small company. At the same time, the reduced scale of the company also made it much easier to approach anybody involved and start a casual conversation about the company's work and my interests.

units of analysis, tasks, and groups of tasks made it hard for the engineer and I to figure out exactly what the other person was talking about. As with the first study, we worked through the same conversations three of four separate times.

Eventually, my engineer friend told me about an entire open source scripting language that had been built into the platform at an earlier point in the development, but that was not included in any of the formal documentation. He thought that I might be able to use the language to make the software do what I wanted. After reading the scripting language documentation and fiddling around with a few more pilot studies, it did in fact turn out that this language could manage the sort of randomization I needed without any customizations.

Unique assignment was harder this time because it undermined some of the assumptions the engineers held about how crowdsourcing tasks tended to work and which they had therefore written into the code. Luckily, I was able to convince them, in the context of a separate discussion with clients interested in using the software to run surveys, that the feature would actually be useful for a wide range of applications and so the lead engineer felt he could justify putting a couple of hours into implementing the new feature late one night.

In hindsight, nothing about either of these experiences – either the one with the developers at the academic research center or the one with the engineers at the startup – strikes me as especially surprising. Perhaps the only part of either story

that really defies explanation is why I ever thought the process would be faster, simpler or less challenging than it turned out. As a social scientist with relatively minimal programming experience I had not anticipated the complexity of software development as a process. I had also failed to explain our needs and concerns in a way that made intuitive sense to software developers, who likewise struggled to explain the inner-workings of their code in a way that a non-developer could easily follow.

Hired Hands Revisited

In the end, both experiments worked out well and the experience helped me apply crowdsourcing tools for the purpose I originally intended: distributed content analysis. I have now completed multiple pilots and small-scale side-projects applying these techniques and am designing several larger studies. When colleagues learn that some of my research involves data collected through Mechanical Turk, they tend to express disbelief that anybody would actually do crowdsourcing work for so little money. They also often voice doubts that the workers on Mturk and other online labor markets answer questions honestly or perform the tasks requested of them in earnest. They never suspect that my early failures with crowdsourced data collection would derive from my own mistakes or

my inability to communicate effectively with software engineers. Such responses overlook the practical challenges of learning how to work with unfamiliar digital tools or the process of software development. They also betray an underlying mistrust of "hired hand research" more generally. While this mistrust is warranted in the context of crowdsourced data, I find it overstated and somewhat ironic. Researchers engaged in crowdsourcing data collection and analysis have developed methods that account for the inevitability of low-quality data with robust techniques to maximize the precision of their results. At the same time, researchers performing more traditional forms of content analysis, interviewing, and survey research have few corresponding solutions to correct for the identical problems.

A little background on crowdsourcing workers provides a useful entry point to a broader discussion of reliability and data quality in the context of crowdsourcing. A diverse population of individuals participate in crowdsourcing as workers in online labor markets or as volunteers in non-commercial projects. For example, while Amazon refuses to make representative data about Mechanical Turk workers publicly available, several informal surveys and qualitative studies provide evidence that the population includes a moderate gender balance as well as individuals from a range of socio-economic, linguistic and cultural backgrounds (Ipeirotis, 2010; Khanna, Ratan, Davis, and Thies, 2010; Ross, Irani, Silberman, Zaldivar, and Tomlinson, 2010). In the experiment I conducted with John Horton

and Daniel Chen, we also found evidence of substantial variation in Internet skills among the participants in our study. Some of this diversity makes Mturk a *more* attractive site for comparative research than traditional laboratories or some other, less diverse online environments. For example, Judd Antin and I found evidence of different motivational patterns across Mturk workers in India when compared with those in the U.S. While it does not make sense to claim that such findings are representative of broader national trends in either country, the experiment incorporates the variations of the site's user population into the research design, taking advantage of the fact that Mturk makes it relatively affordable to run a cross-national research project.

The diversity of Mturk workers creates some interesting challenges in the day-to-day process of running a study. Earlier, I alluded to the fact that workers can email you about any task that you post to the site. In that case, a worker alerted me to a problem with one of my tasks, saving me some additional frustration and research funds. In the course of piloting and running the rest of that same experiment, I received close to 100 other emails from workers. Most of them were not nearly so helpful, polite, or grammatically coherent. Many reported problems viewing my task (the majority of these were caused by old versions of Internet Explorer failing to render the site correctly) or simply requested that I submit payments for the task more quickly (I tended to process the payments in monthly

batches, which was nowhere near fast enough for some participants). I setup auto-responses and stock replies to common inquires in order to alleviate the inbox traffic, but I had effectively become the payroll and technical support department of a very small, trans-national research organization overnight. Somehow, I had wound up at both ends of an outsourcing supply chain at the same time.

In terms of assessing the validity and reliability of data, the diversity of skills, languages, and educational backgrounds among Mturk workers has given rise to some creative solutions that improve upon traditional content analysis techniques. Historically, content analysis research has employed a multistage process to ensure that the results are both valid and reliable (see Krippendorff, 2004; Neuendorf, 2002). The basic idea behind reliability in this context is that you want to be sure that your results are not simply the product of an idiosyncratic interpretation of the world, but rather correspond to a widely intelligible and acceptable interpretation. Methodologically, the state of the art way to test this is pretty simple: you have multiple individuals apply the content analysis instrument (or codes) to a bunch of the same content (text, images, videos, or whatever) and then run some statistical tests on the results to calculate the rate of agreement between the coders.

Nothing about crowdsourcing changes the underlying objective here: you still want your results to capture real concepts and you still need to get some sort of

confirmation that these concepts hold up at an inter-subjective level. The core difference just lies in process by which you have to go about measure reliability. The problem with the standard approaches to reliability in the context of crowdsourcing is severalfold. Unlike a more typical research project employing undergraduate research assistants, crowdsourcing, by definition, opens up the project to a wider population of people about whom you, the researcher, know pretty much nothing and over whom you have very little influence. In addition, the scale of participation possible with crowdsourcing makes it feasible for you to incorporate results from even more coders about even more questions than a traditional content analysis study. In this sense, Roth's aforementioned concerns about "dubious guesses" take on a new urgency. The messy data generated by a crowd is, by definition, more dubious than most.

Luckily, a lot of smart statisticians and computer scientists have come up with creative responses to processing unreliable data from multiple coders.[14] The key involves approaching the raw, messy data as an optimization problem: for every data point we want to maximize the likelihood that we select the correct answer on

---

[14] Everything I am about to say comes directly from previous research in Maximum Likelihood Estimation and human computation research (Dawid and Skene, 1979; Sheng et al., 2008; Snow et al. 2008).

the basis of the coders' responses. We (or more accurately, our statistical software), can then treat each coder as possessing an underlying probability of providing a correct response to any question, and can estimate this probability for each coder by asking them to answer a few questions to which we know the answer ahead of time. After establishing the workers' rate of accuracy, we collect multiple responses to every question from multiple coders and weight these observations by the coder's accuracy rate to estimate the answer that is most likely to be correct.

The elegance of this approach stems from the use of "bad" information to produce valid, reliable inferences about the best answers to any particular question. In Roth's terms, you take a lot of dubious guesses, try to figure out how dubious they are, and then use that knowledge to estimate the least dubious guess. Instead of fighting or denying the hired hands problem, this technique takes advantage of it in a clever way.

No Silver Bullets

If you step back from the details, the sort of algorithmic approaches to maximizing data quality contain more general implications for research design and methods. As I said in the introduction, I think all research work entails a process of combining tools and perspectives in an effort to extract accurate and reliable knowledge about a particular topic. The extent to which this process involves hired hands, online

participants, collaborators unfamiliar with social scientific methods, or computational techniques varies from project to project. In the present case, all of these factors introduced sources of error, bias, and uncertainty into my work. As with the failed pilots and bad-quality data that came from some of the respondents in my Mturk studies, I think these sorts of challenges constitute an important part of the learning process that eventually allowed me to produce better studies. In retrospect, it does not make sense to me to even pretend that I could have completely eliminated all the mistakes along the way. Instead, the best I could do was to incorporate these mistakes into my knowledge of the process and try to optimize the process within the limits of that knowledge.

If you think of research as a constructive learning process involving multiple people, procedures, and tools, each of which introduces some sort of bias and error along the way, I think the question of hired hands opens up a broader discussion. The big point here is that any kind of wrong information is still informative – whether about the quality of the data, the reliability of the data collection processes, or the nature of the people participating in the research project. As a result, every dubious guess provides an opportunity to learn, to refine and improve the research project and the finished product it generates.

In this sense, I think there comes a time in any research project when each of us, intentionally or not, undermines the validity, reliability, or accuracy of the work.

As researchers, we may do this by imposing an overly idealized vision of the research process on the reality of data collection (that tends to be messy, unpredictable, and biased in some way or another). We may also do this by naively expecting our research assistants and collaborators to share our assumptions, objectives and desires.

For classic “hired hands” like Roth's research assistants or the typical worker on Mechanical Turk, the reasons behind this have to do with the structural constraints imposed by the division of labor and the dynamics of paid work in labor markets. In these markets, the exigencies of wage earning too often result in people doing a task they find uninspiring, boring or stupid. In the context of Mechanical Turk and other similar online platforms, these dynamics tend to be exacerbated by extraordinarily atomized and anti-social work environments that do little to counter-balance the intrinsic inequalities between workers and employers.

For collaborators like my software engineer friends, the story gets more complicated: the ways in which they undermined the project's progress and objectives did not seem to stem from the desire to shirk responsibility for financial or other reasons. Instead, there was an underlying misunderstanding that simply took a lot of work and communication to correct. Once everybody was on the same page, they quickly and happily helped me execute randomization in both of my studies.

Needless to say, I am still quite far removed from the vision of crowdsourced data collection that my colleagues and I imagined around that conference table. In fact, almost three years later, I am just beginning to realize some of the plans we laid out. It turns out that even when you can do research on and about a networked environment where the cost of attracting thousands of people to participate in your study approaches zero, nobody and no technology will ever truly do all the work for you.

References

Antin, J., and A. Shaw. 2011. "Social Desirability Bias and Self-Reports of Motivation: A Cross-Cultural Study of Amazon Mechanical Turk in the US and India." Presented at the 2011 ACM Confernce on Computer-Supported Cooperative Work (CSCW2011). Hangzhou, China.

Benkler, Yochai. 2006. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. New Haven: Yale University Press.

Brooks, Frederick P. 1995. *The Mythical Man-Month: Essays on Software Engineering*. Anniversary ed. Reading, MA: Addison-Wesley.

Dawid, A. P., and A. M. Skene. 1979. "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm." *Journal of the Royal*

Statistical Society. Series C (Applied Statistics) 28(1):20-28. Retrieved May 19, 2011.

Fisher, Ronald Aylmer. 1935. *The Design of Experiments*. Edinburgh, Scotland: Oliver and Boyd.

Grier, David Alan. 2005. *When Computers Were Human*. Princeton: Princeton University Press.

Holland, Paul W. 1986. "Statistics and Causal Inference." *Journal of the American Statistical Association* 81(396):945-960.

Horton, John J, David Rand, and Richard J Zeckhauser. 2010. "The Online Laboratory: Conducting Experiments in a Real Labor Market." *Experimental Economics*.

Howe, Jeff. 2008. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Crown Business.

Ipeirotis, P. 2010. "Demographics of Mechanical Turk." *New York University Working Paper*.

Kanefsky, B., N. G. Barlow, and V. C. Gulick. 2001. "Can Distributed Volunteers Accomplish Massive Data Analysis Tasks?" *Proceedings of the Lunar and Planetary Institute Science Conference* 32:1272. Retrieved June 1, 2011.

Khanna, Shashank, Aishwarya Ratan, James Davis, and William Thies. 2010. "Evaluating and improving the usability of Mechanical Turk for

low-income workers in India." Pp. 12:1–12:10 in *Proceedings of the First ACM Symposium on Computing for Development*, *ACM DEV '10*. London, United Kingdom: ACM.

Krippendorff, Klauss. 2004. *Content Analysis: An Introduction to Its Methodology*. 2nd ed. Thousand Oaks, CA: Sage.

Little, Roderick J., and Donald B. Rubin. 2000. "Causal Effects in Clinical and Epidemiological Studies Via Potential Outcomes: Concepts and Analytical Approaches." *Annual Review of Public Health* 21:121-145.

Neuendorf, Kimberly A. 2002. *The Content Analysis Guidebook*. SAGE.

Ross, Joel, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson. 2010. "Who are the crowdworkers?: shifting demographics in mechanical turk." Pp. 2863–2872 in *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*. Atlanta, Georgia, USA.

Roth, Julius A. 1966. "Hired Hand Research." *The American Sociologist* 1(4):190-196.

Shaw, Aaron, John J. Horton, and Daniel L. Chen. 2011. "Designing Incentives for Inexpert Human Raters." *Proceedings of the 2011 ACM Conference on Computer-Supported Cooperative Work (CSCW2011)* Hangzhou, China.

Sheng, Victor S, Foster Provost, and Panagiotis G Ipeirotis. 2008. "Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers." *Proceedings of the Conference on Knowledge Discovery and Data Minding 2008 (KDD-2008)*.

Snow, Rion, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. "Cheap and Fast—but is it good? Evaluating Non-expert Annotations for Natural Language Tasks." *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*.

Williams, Dmitri, and Li Xiong. 2009. "Herding Cats Online: Real Studies of Virtual Communities." Pp. 122-140 in *Research Confidential: Solutions to Problems Most Social Scientists Pretend They Never Have*, edited by Eszter Hargittai. Ann Arbor, MI: University of Michigan Press.

Winner, Langdon. 1986. "Do Artifacts Have Politics?" Pp. 19-39 in *The Whale and the Reactor: A Search for Limits in an Age of High Technology*. Chicago: University of Chicago Press.