Why I Hate Mechanical Turk Research (and Workshops)

Eytan Adar
University of Michigan
105 S. State St., Ann Arbor, MI 48104
eadar@umich.edu

ABSTRACT

There is a certain enthusiasm in any community when encountering a new tool—a shiny new instrument that appears to solve hard problems—that leads to a barrage of research "results." However, by rewarding quick demonstrations of the tool's use, we fail to attain a deeper understanding of the problems to which it is applied, to pursue real solutions rather than stopgaps, or to develop a real understanding of the tool's limits. In this paper I argue that we are currently experiencing these failures in our focus within crowdsourcing (both *crowdsourced science* and the *science of crowdsourcing*) but that there are still some interesting research trajectories available to us. They just might require significant work and produce the most dreaded of research outcomes: negative results.

INTRODUCTION

Part of the reason in making this argument was a growing frustration in the number of papers and projects—the research energy of the community—in targeting low hanging fruit in the crowdsourcing domain (by no means all papers, but a growing number of them). The exuberance in which such projects are pursued may be an inevitable part of any new scientific or engineering development, but it is not apparent to me that it is actually healthy. When a community decides to reward such efforts, rather than "deeper" work, I believe it is doing a disservice to itself. Eventually, of course, we'll start to recognize this behavior and will become more critical, but I would like to argue that we should do this sooner rather than later.

When I first wrote about this, I titled the blog post, "Why I Hate Mechanical Turk Research." [1] (a tongue-in-cheek title, as is this paper) and received a lot of comments, both agreeing and disagreeing with what I wrote. Using this feedback, and some additional thinking, I'll try to make a better argument for my objections here to address some of the issues that were brought up.

The bulk of crowdsourcing papers that have appeared fall into two categories: *crowdsourced science* and the *science of crowdsourcing*. I'll try to address my criticisms of each individually before getting to the combinations (oftentimes the worst of both). Unlike the blog post, I'll also try to offer some directions that I think are worth exploring. They're not easy—and this likely makes them less attractive—but I hope some will be pursued.

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

DISCLAIMER

I have been told by many who have read the blog post that it's not entirely obvious what I don't like. While it's easy to point to papers that I think are good (see the references for examples), I'm still uncomfortable pointing at specific papers I think are bad. There's a danger in doing it this way. I suspect every reader will identify with some "good" paper ("oh, my paper is just like this other one."). I nonetheless hope the ideas here will at least encourage discussions that allow us to move forward instead of spinning our wheels.

CROWDSOURCED SCIENCE

As I suggest in [1], I have no issues with *some* kinds of crowdsourced/crowdsourcing research. For example:

1. Work where the *research product* is understanding humans or human interactions (e.g., a psychology or behavioral economics experiment),
2. finding novel ways of breaking up complex problems into things that zero/marginal-expertise agents *can* and *want* to do (e.g., FoldIt [6] or *specific* game-with-a-purpose instances [22])

I have no quibbles with (1). The product of such research is theory or some other downstream application. We have a demand for people, systems like MTurk have a supply, and if we can convince ourselves that they are the right kind of people, who will do the task we want, we have a match.

I also have a lot of appreciation for (2). Getting non-biologists to perform protein folding is impressive [6]. It means breaking up the task into something that is both rewarding, doable, and that the activity produces some kind of interesting and/or useful product. This kind of work is all the more impressive when the crowdsourced results are *significantly* better/faster than existing computational means. That said, we should not fool ourselves into believing that all hard problems fit this mold or completely distract ourselves from advancing other, computational means of solving these problems. More importantly, we

should not fool ourselves into believing that we have done something new by using human labor.

Humans doing human work

Showing that humans can do human work is not a contribution. I maintain that one of the worst trends in crowdsourced science work is demonstrating the obvious. Specifically, many results have the pattern of identifying a particular problem that is hard computationally, but extremely easy for humans, and then "solving" it through crowdsourcing. We know that most people can do this kind of work (find a car in a picture, pick out cancer cells, label sentences). The surprise would be if Mechanical Turk Workers couldn't.

The other problem I see in many papers is that the computational "baseline" is sufficiently good such that any improvements through the use of crowd work are not particularly impressive. These baselines, however, are simply ignored—treated as is if they never existed.

Even when performance comparisons are made (e.g., NLP task X is 20% better through crowdsourcing), frequently scaling arguments are missed. A 20% improvement is not interesting when we can only get it on less than 1% of the dataset due to lack of workers or funding[1].

When I pointed out in a review that a hard, critical, large-scale bootstrapping step had been glossed over, the authors countered with a small experiment that indicated that Amazon Mechanical Turk Workers (i.e., Turkers) could be paid 4 cents per task. QED. Proof-by-Turk-existence. The reality that they decided to ignore was that they could never get enough Turkers to actually bootstrap the system and that the costs of the approach might easily cost millions.

Attention Economics

This is not to argue that computational solutions magically work at the speed and scale or at the cost we would desire. However, we are frequently able to make reasonable arguments about current technology trends that will allow us to catch up ("proof-by-Moore's law"). Although not always satisfying, we have some evidence that this is true (or at least some data to argue one way or the other).

With crowdsourcing, it is hard to imagine a world in which there is this kind of scaling—where costs per "cycle" decreases significantly and/or where the number of workers grows. Sure, we might have rapid population growths, but people are still an expensive, limited resource that can only split their time so many ways. Can we really expect enough global human attention to meet the demands of the workloads we are imagining? If the Turk-like systems are truly a success, the demand for (good) workers will outpace the supply inevitably translating to higher costs. Can you still have your task completed for $0.05 when that happens?

[1] I'm not insensitive to the argument that only a small fraction of tasks (that < 1%) require human intervention, but I think these arguments are rarely made convincingly.

THE SCIENCE OF CROWDSOURCING

In contrast to crowdsourced science, the science of crowdsourcing targets the systems themselves, where the applications—if they exist—are largely secondary:

3. writing the user's manual for MTurk/crowdsourcing research [9, 10, 13, 16, 17]
4. figuring out how to make crowdsourcing more/better/faster (i.e., more use, better signal, faster response) [3, 5, 7, 11, 15, 19, 21]

I think some amount of (3) is necessary, in part because it helps to have use cases of good and bad experiences with crowdsourced jobs, and to convince ourselves that crowdsourced science is OK. Showing, for example, that Turkers are "the same as everybody else" (i.e., college undergraduates) requires some demographic analysis and replication of classic experiments. Demonstrating, once or twice, that Turkers are capable of doing work in some *specific* domain (e.g., NLP [20] or vision) feels appropriate. Maybe. Some, and I don't know how much, of this type of work is necessary but there is probably more of it being done than we need. From the evidence thus far, I remain somewhat skeptical that Turkers are some strange beast we've never encountered before. Like many other people we might use for labor: Turkers are human, Turkers are unreliable and Turkers maximize "profits."

Those doing research of type (4) struggle against the limits of the platform, seeking to squeeze out a better signal from crowdsourced work. Having identified situations in which the simple model does not address a class of problems, researchers have rushed in to fix these by making crowdsourcing iterative, synchronous, (near) real-time, dynamic resource allocation, or integrated with other computational infrastructures (i.e., computational machines with an Oracle). Alternatively, researchers have developed mechanisms for noise removal that either try to get individuals to produce a better signal, or extracting a better signal from the combination of workers.

Others have identified the "group" as the target of opportunity for research. It is true that in many situations getting a *group*—especially a loosely controlled one—to do a task is a hard problem. That is, while the individual is easy to control, the group is not. While this is true, it is not obvious how much of this is a problem specific to Turk-like frameworks or that the problems are realistic. If we can make a group of Turkers do the right thing, the argument goes, we have evidence that the prototype system will translate into the real world. I'm not sure I buy this reasoning. Much of what is hard in getting Turkers to behave (e.g., not having strong reputation or reliability models) goes away in real systems (to be replaced by other hard problems, such as long-term consistency of work).

Some of this work has value—it enables many design patterns that a clever domain scientist might apply. The question for the community, is at which point at which the science of crowdsourcing stop being interesting research? At some point the solutions become more and more fanciful, demonstrating how smart we are, but not really solving a real problem. I'm not objecting to this on principle. I can see the attraction to work that shows off our cleverness. I simply believe that the community can be more honest about this fact and what that means to users of these techniques (e.g., more like [11]).

Lessons from Other Disciplines

In building the science of crowdsourcing it is also very easy, and cheap, to reinvent the wheel. Instead, we should (though we rarely do) turn to other scientific disciplines. Von Neumann taught us about majority voting, and the hundreds of system architecture and OS researchers that followed him only added to this literature (voting, RAID, parallel development, etc.). Similarly, "manipulating" and "incentivizing" people to behave better (making the component less noisy) are also thoroughly addressed in psychology, business, economics, statistics, machine learning, survey design, and marketing research (they just call it different things: zero/marginal-intelligence agents, information markets, learning from multiple experts, etc.) We should be careful not to over-claim metaphorical connections, but there's a great deal of related work in other fields that we should leverage through thoughtful survey articles and tutorials. Mostly, we should remember that new to us does not mean new.

AT THE INTERSECTION

One dangerous trend is the attempt to simultaneously do crowdsourced science while doing the science of crowdsourcing. Mostly, I think the "hard" parts—those pieces of research that actually create advancement—are not the same in the two types of research, and that it is too difficult to get both right.

In many situations, the crowdsourced solution to the scientific application is so specific that any lessons that can be drawn are not generalizable (the way in which the tool is used is too unique). In other cases, the crowdsourcing tool is being used in such a standard way, that it should just be reported as a part of the methodology. Forcing application researchers to over-emphasize design implications or generalizations detracts from the main work.

Conversely, papers that might have a real contribution in the science of crowdsourcing frequently over-emphasize one specific, simple, application. It is not the "simple" that I object to, rather it's the "one." Such contributions should concentrate on being broadly applicable and demonstrated on multiple applications. By necessity, we frequently resort to "toy" applications for this demonstration. This should not be considered a negative: a well-planned set of these makes it possible to identify the important contributions of the new crowdsourcing technique.

Some may be able to pull off both kinds of contributions, but for most, concentrating on one type is more interesting and useful.

DOING IT DIFFERENTLY

The optimism around Crowdsourcing will probably at some point be tempered with certain realities. The Turk, and systems like it, will likely be relegated to the same status as the economists' z-Tree [8]—tools to be drawn upon for running certain experiments. The amount of noise, the difficulty in breaking down interesting problems into sub-tasks, our inability to conduct certain kinds of experiments at all, creates limits to the applicability of the tool. Some, but not all, of these limits will be met with newer and better crowdsourcing tools. Other challenges such as spam and poor worker quality will likely drive tools to include features for tracking identity and reputations. It's at these eventualities that there are a number of interesting questions that are worth pursuing (even if they are negative results):

- **Reputation**: Given the addition of reputation and identity, how does the market price a "high reputation" individual? How much more will a Turker cost?
- **Adversaries**: With higher market prices do we begin to see more insidious "adversaries?" What is a realistic and appropriate adversary model? Do all our statistical/AI techniques break under these threats?
- **Work for nothing**: Can we break out of the attention economics trap and identify nonreactive or parasitic [2] ways to leverage human effort that don't require people to "work" within a given framework?
- **Oracles**: Given likely changes in cost, can we make good engineering decisions in identifying the conditions when a human oracle(s) should be consulted? [4, 18]
- **Creativity**: What computational tasks cannot be broken apart? Which human tasks cannot be broken apart? For example, can we have Turkers produce creative or aesthetically pleasing products or do we end up with "camels?"[2]

These questions are by no means exhaustive (or necessarily interesting or good) but I think are of the general class of research that will allow us to capture the strengths *and* weaknesses of crowd-labor systems and make them a useful piece of our toolkit.

[2] "A camel is a horse designed by committee" – Sir Alec Issigonis

CONCLUSIONS

There are still a number of contributions to be made *through* crowdsourcing and *to* crowdsourcing. However, I believe that an over-optimistic perspective on what crowdsourcing might offer is detrimental. It leads us to stop questioning the structure and conclusions of crowdsourced/crowdsourcing research. Above, I outlined classes of research work involving crowds. Each suffers from limits that we are actively ignoring. I believe that not only should these limits be addressed, but they should be embraced as interesting research questions in their own right. The payoff will be a tool we can use correctly.

ACKNOWLEDGMENTS

I suspect many people would prefer to not be listed here. These individuals may not have agreed with my ideas (in fact some probably objected vehemently), but they were still gracious enough to listen and provide feedback. So thanks to Michael Bernstein, Jeffery Bigham, Rob Miller, Paul Resnick, Mark Ackerman, Sean Munson, Michael Cafarella, Lada Adamic, Erin Krupka, Dan Weld, Jessica Hullman, Michael Toomim, and Panos Ipeirotis the MISC group, and the readers and commenters on my blog.

REFERENCES

1. Eytan Adar, "Why I Hate Mechanical Turk Research," Nov. 11, 2010, http://blog.cond.org/?p=28
2. A.-L. Barabási, V. W. Freeh, H. Jeong, J. Brockman. 2001. Parasitic computing, *Nature* 412, 894-897
3. Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. UIST '10
4. Michael Bernstein and Inbal Talgam-Cohen. 2010. Human computation and crowdsourcing. *XRDS* 17(2):6
5. Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and Tom Yeh. 2010. VizWiz: nearly real-time answers to visual questions. UIST '10.
6. Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović & Foldit players. 2010. Predicting protein structures with a multiplayer online game, *Nature* 466, 756–760
7. Peng Dai, Mausam, and Daniel Weld. 2010. Decision-Theoretic Control for Crowdsourced Workflows, AAAI'10.
8. Urs Fischbacher. 2010. "z-Tree: Zurich toolbox for ready-made economic experiments," *Experimental Economics*, 10(2):171-178.
9. John Joseph Horton and Lydia B. Chilton. 2010. The labor economics of paid crowdsourcing. EC '10.
10. Panagiotis G. Ipeirotis, 2010 "Demographics of the Mechanical Turk," Working Paper
11. Panagiotis G. Ipeirotis, "The unreasonable effectiveness of simplicity," February 6, 2011, http://behind-the-enemy-lines.blogspot.com/2011/02/unreasonable-effectiveness-of.html
12. P. Ipeirotis, F. Provost, V. Sheng, and J. Wang. 2010. Repeated Labeling Using Multiple, Noisy Labelers, Working Paper
13. Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with Mechanical Turk. CHI'08.
14. Gunnar W. Klau, Neal Lesh, Joe Marks, Michael Mitzenmacher, and Guy T. Schafer. 2002. The HuGS platform: a toolkit for interactive optimization. AVI '02.
15. Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. 2010. TurKit: human computation algorithms on mechanical turk. UIST '10.
16. Mason, W., & Watts, D. J. 2009. Financial Incentives and the "Performance of Crowds". HCOMP'09.
17. G. Paolacci, J. Chandler, P. Ipeirotis. 2010. Running Experiments on Amazon Mechanical Turk, *Judgment and Decision Making*, Vol. 5, No. 5
18. Dafna Shahaf and Eyal Amir, 2007. Towards a Theory of AI-Completeness., CommonSense '07.
19. Dafna Shahaf and Eric Horvitz. 2010. Generalized task markets for human and machine computation., AAAI'10.
20. Rion Snow, 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. *EMNLP '08*.
21. Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. 2010. The Multidimensional Wisdom of Crowds, NIPS'10.
22. Luis von Ahn. 2006. Games with a Purpose. *Computer* 39(6):92-94