

Prototype Tasks: Improving Crowdsourcing Results through Rapid, Iterative Task Design

Snehalkumar ‘Neil’ S. Gaikwad, Nalin Chhibber, Vibhor Sehgal, Aipta Ballav, Catherine Mullings, Ahmed Nasser, Angela Richmond-Fuller, Aaron Gilbee, Dilrukshi Gamage, Mark Whiting, Sharon Zhou, Sekandar Matin, Senadhipathige Niranga, Shirish Goyal, Dinesh Majeti, Preethi Srinivas, Adam Ginzberg, Kamila Mananova, Karolina Ziulkoski, Jeff Regino, Tejas Sarma, Akshansh Sinha, Abhratanu Paul, Christopher Diemert, Mahesh Murag, William Dai, Manoj Pandey, Rajan Vaish and Michael Bernstein

Stanford Crowd Research Collective
daemon@stanford.edu

Abstract

Low-quality results have been a long-standing problem on microtask crowdsourcing platforms, driving away requesters and justifying low wages for workers. To date, workers have been blamed for low-quality results: they are said to make as little effort as possible, do not pay attention to detail, and lack expertise. In this paper, we hypothesize that *requesters* may also be responsible for low-quality work: they launch unclear task designs that confuse even earnest workers, under-specify edge cases, and neglect to include examples. We introduce *prototype tasks*, a crowdsourcing strategy requiring all new task designs to launch a small number of sample tasks. Workers attempt these tasks and leave feedback, enabling the requester to iterate on the design before publishing it. We report a field experiment in which tasks that underwent prototype task iteration produced higher-quality work results than the original task designs. With this research, we suggest that a simple and rapid iteration cycle can improve crowd work, and we provide empirical evidence that requester “quality” directly impacts result quality.

Introduction

Quality control has been a long-standing concern for microtask crowdsourcing platforms such as Amazon Mechanical Turk (AMT) (Ipeirotis, Provost, and Wang 2010; Marcus and Parameswaran 2015). Low quality work causes requesters to mistrust the data they acquire (Kittur, Chi, and Suh 2008), offer low payments (Ipeirotis 2010), and even leave the platform in frustration. Currently, the dominant narrative is that low-quality work is the fault of *workers*—those who demonstrate “a lack of expertise, dedication [or] interest” (Sheng, Provost, and Ipeirotis 2008), or are either “lazy” or “overeager” (Bernstein et al. 2010). Workers are blamed for cheating (Christoforaki and Ipeirotis 2014; Martin et al. 2014), putting in minimal effort (Bernstein et al. 2010; Rzeszotarski and Kittur 2011), and strategically maximizing earnings (Kittur, Chi, and Suh 2008). To counter such behavior, requesters introduce worker-filtering strategies such as peer review (Dow et al. 2012), attention checks with known answers (Le et al. 2010), and majority agreement (Von Ahn and Dabbish 2004). Despite these interventions, the quality of crowd work often remains low (Kittur et al. 2013).

Copyright © 2017, HCOMP, Association for the Advancement of Artificial Intelligence (www.aaai.org), Authors reserve all rights.

This narrative ignores a second party who is responsible for microtask work: the *requesters* who design the tasks. Since HCOMP’s inception in 2009, many papers have focused on quality control by measuring and managing crowd workers — however, very few have implicated requesters and their task designs in quality control. For example, a system might give workers feedback on their submission, while taking the requesters tasks as perfect and fixed (Dow et al. 2012). This assumption is even baked into worker tools such as Turkooption (Irani and Silberman 2013), which include ratings for fairness and payment but not task clarity. *Should requesters also bear responsibility for low-quality results?*

Human-computer interaction has long held that errors are the fault of the designer, not the user. Through this lens, the designer (requester) is responsible for conveying the correct mental model to the user (worker). There is evidence that requesters can fail at this goal, forgetting to specify edge cases clearly (Martin et al. 2014) and writing text that is not easily understood by workers (Khanna et al. 2010). Requesters, as domain experts, underestimate task difficulty (Hinds 1999) and make a fundamental attribution error (Ross 1977) of blaming the person for low-quality submissions rather than the task design that led them to produce those results. Recognizing this, workers become risk-averse and avoid requesters who could damage their reputation and income (Martin et al. 2014; McInnis et al. 2016).

Workers and requesters share neither the same expectations for results nor their objectives for tasks (Kittur, Chi, and Suh 2008; Martin et al. 2014). For instance, requesters may feel that the task is clear, but that workers do not put in sufficient effort (Bernstein et al. 2010; Kittur, Chi, and Suh 2008), thus rejecting work that they judge to be of low quality. Workers, on the other hand, may feel that requesters do not write clear tasks (McInnis et al. 2016) or offer fair compensation (Martin et al. 2014). Overall, requesters’ mistakes lead workers to avoid tasks that appear too confusing or ambiguous in order to avoid getting rejected and harming their reputation (Irani and Silberman 2013; McInnis et al. 2016).

Rather than improving result quality by designing crowdsourcing processes for workers (McInnis et al. 2016; Dow et al. 2012), in this paper we capitalize on the notion of broken windows theory (Wilson and Kelling 1982) and improve quality by designing crowdsourcing processes for re-

requesters. Our research contributes: 1) an integration of traditional human-computer interaction techniques into crowdsourcing task design via prototype tasks, and, through an evaluation, 2) empirical evidence that requester “quality” directly impacts task quality.

Prototype Task System

While prior work has studied the impact of enforcement mechanisms such as attention checks (Kittur, Chi, and Suh 2008; Mitra, Hutto, and Gilbert 2015), our focus is on requesters’ ability to build common ground (Clark and Brennan 1991) with workers through instructions, examples, and task design. Inspired by the iterative design process in human-computer interaction (Preece et al. 1994; Sharp, Jenny, and Rogers 2007; Harper et al. 2008; Holtzblatt, Wendell, and Wood 2004), we incorporate worker feedback in the task design process. We build on the prototyping and evaluation phases of user-centered design (Baecker 2014; Dix 2009; Holtzblatt, Wendell, and Wood 2004; Nielsen 2000) and introduce *prototype tasks* as a strategy for crowdsourcing task design (*Appx. B*). Prototype tasks, in-progress task design, draw on an effective practice in crowdsourcing of identifying high-quality workers by posting small tasks to vet them first (Mitra, Hutto, and Gilbert 2015; Stephane and Morgan 2015).

Prototype tasks, integrated into the Daemo crowdsourcing marketplace (Gaikwad et al. 2015), require that all new tasks go through a feedback iteration from 3–7 workers on a small set of inputs before being launched in the marketplace. These workers attempt the task and leave feedback on the task design. First, prototype tasks involve adding a feedback form to the bottom of the task. This might be a single textbox, or following recent design thinking feedback practice (d.school 2016), a trio of “I like...”, “I wish...” and “What if...?” feedback boxes. Second, to launch a prototype task, a requester samples a small number of inputs (e.g., a few restaurants to label), and launches them to a small number of workers. Third, they utilize the results and feedback to iterate on their task design (*Appx. B*).

We argue that prototype tasks, which take only a few minutes and provide a “thin slice” of feedback from a few workers, are enough to identify edge cases, clarify instructions, and engage in perspective-taking as a worker. *We hypothesize that this iteration will result in better designed tasks, and downstream, better results.*

Evaluation and Results

Prototype tasks embody a proposition that requesters task design skills directly impact the work quality they receive, and that processes from user-centered design might produce higher-quality task designs. To investigate the claim, we recruited 7 experienced microtask crowdsourcing requesters. Requesters each designed three different canonical types of prototype tasks detailed in prior work (Cheng, Teevan, and Bernstein 2015), then launched them to a small sample of workers on AMT for feedback, and reviewed the results to iterate and produce revised task designs. The types of the tasks were chosen based on the common crowdsourcing microtask categories: categorization, search, and description

(*Appx. A*). Each task was known to be either susceptible to misinterpretation or a challenging edge case. We provided requesters with a short description of the tasks they should implement and private ground-truth results that they should replicate. We then launched both the original and the revised tasks on AMT, randomizing workers between the two task designs. Requesters reviewed pairs of results blind to condition and chose the set they found to be higher quality. If workers are primarily responsible for low-quality crowd work, then the original and revised tasks would both have equally poor results. However, if the poor quality is due at least in part to the requesters, then the prototype tasks would lead to higher-quality results.

We found that the prototype task intervention produced results that were preferred by requesters significantly more often than the original design. Requesters preferred the results from the Prototype (revised) tasks nearly two thirds of the time (51 vs. 31). This result was significant ($\chi^2(1)=4.878, p < 0.05$). In other words, modifications to the original task design and instructions had a positive impact on the final outcome. Thus, our hypothesis was supported: result quality is at least in part attributable to requesters’ task designs. The requesters themselves did not change, and the distribution of worker quality was identical due to randomization, but improved task designs led to higher-quality results. While only seven participants limits generalizability, the significant result with just seven is an indirect statement of the large effect size of the intervention. Future work involving the longitudinal experiments of tasks will help better triangulate the phenomenon and dynamics of the requesters’ and workers’ impact on quality.

To understand where requesters might misstep, we next investigated the written feedback that workers gave during the first phase, and the changes that requesters made to their tasks in response to the feedback. We conducted a grounded analysis using workers’ feedback on flaws in tasks’ designs and instructions. Two researchers independently coded each item of feedback and each task’s design changes according to inductively derived categories from data, resolving disagreements through discussion. Raw Cohen’s κ agreement scores for these categories ranged from 0.64–1.0, indicating strong agreement. This process generated four categories of feedback and iteration: unclear/vague instructions, missing examples, incorrect grammar, and broken task UI/accessibility (*Appx. B*). The most common worker feedback was on the user interface and accessibility of the task (49% of feedback).

Conclusion

In this paper, we introduce prototype tasks, a strategy for improving crowdsourcing task results by gathering feedback on a small sample of the work before launching. We offer empirical evidence that requesters’ task design directly impacts result quality. Based on this result, we encourage requesters to take ownership of the negative effects they may have on their own results. This work ensures that the discussion of crowdsourcing quality continues to include requesters’ behaviors and emphasize how their upstream decisions can have downstream impacts on work quality.

References

- Baecker, R. M. 2014. *Readings in Human-Computer Interaction: toward the year 2000*. Morgan Kaufmann.
- Bernstein, M. S.; Little, G.; Miller, R. C.; Hartmann, B.; Ackerman, M. S.; Karger, D. R.; Crowell, D.; and Panovich, K. 2010. Soylent: A word processor with a crowd inside. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, 313–322. New York, NY, USA: ACM.
- Bernstein, M. S.; Teevan, J.; Dumais, S.; Liebling, D.; and Horvitz, E. 2012. Direct answers for search queries in the long tail. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 237–246. ACM.
- Callison-Burch, C., and Dredze, M. 2010. Creating speech and language data with amazon’s mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, 1–12. Association for Computational Linguistics.
- Cheng, J.; Teevan, J.; and Bernstein, M. S. 2015. Measuring crowdsourcing effort with error-time curves. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 205–221. ACM.
- Christoforaki, M., and Ipeirotis, P. 2014. Step: A scalable testing and evaluation platform. In *Second AAAI Conference on Human Computation and Crowdsourcing*.
- Clark, H. H., and Brennan, S. E. 1991. Grounding in communication. *Perspectives on socially shared cognition* 13(1991):127–149.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR '09*.
- Dix, A. 2009. *Human-computer interaction*. Springer.
- Dow, S.; Kulkarni, A.; Klemmer, S.; and Hartmann, B. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, 1013–1022. ACM.
- d.school, S. 2016. Design method: I like, i wish, what if. <http://dschool.stanford.edu/wp-content/themes/dschool/method-cards/i-like-i-wish-what-if.pdf>.
- Gaikwad, S. N.; Morina, D.; Nistala, R.; Agarwal, M.; Cossette, A.; Bhanu, R.; Savage, S.; Narwal, V.; Rajpal, K.; Regino, J.; et al. 2015. Daemo: A self-governed crowdsourcing marketplace. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 101–102. ACM.
- Harper, E. R.; Rodden, T.; Rogers, Y.; Sellen, A.; Human, B.; et al. 2008. Human-computer interaction in the year 2020.
- Hinds, P. J. 1999. The curse of expertise: The effects of expertise and debiasing methods on prediction of novice performance. *Journal of Experimental Psychology: Applied* 5(2):205.
- Holtzblatt, K.; Wendell, J. B.; and Wood, S. 2004. *Rapid contextual design: a how-to guide to key techniques for user-centered design*. Elsevier.
- Ipeirotis, P. G.; Provost, F.; and Wang, J. 2010. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, 64–67. ACM.
- Ipeirotis, P. G. 2010. Analyzing the amazon mechanical turk marketplace. In *XRDS: Crossroads, The ACM Magazine for Students - Comp-YOU-Ter*, 16–21. ACM.
- Irani, L. C., and Silberman, M. 2013. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 611–620. ACM.
- Khanna, S.; Ratan, A.; Davis, J.; and Thies, W. 2010. Evaluating and improving the usability of mechanical turk for low-income workers in india. In *Proceedings of the first ACM symposium on computing for development*, 12. ACM.
- Kim, J., and Monroy-Hernandez, A. 2016. Storia: Summarizing social media content based on narrative theory using crowdsourcing. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 1018–1027. ACM.
- Kittur, A.; Nickerson, J. V.; Bernstein, M.; Gerber, E.; Shaw, A.; Zimmerman, J.; Lease, M.; and Horton, J. 2013. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, 1301–1318. ACM.
- Kittur, A.; Chi, E. H.; and Suh, B. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 453–456. ACM.
- Lasecki, W. S.; Kushalnagar, R.; and Bigham, J. P. 2014. Legion scribe: real-time captioning by non-experts. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*, 303–304. ACM.
- Le, J.; Edmonds, A.; Hester, V.; and Biewald, L. 2010. Ensuring quality in crowdsourced search relevance evaluation. In *SIGIR workshop on crowdsourcing for search eval*.
- Marcus, A., and Parameswaran, A. 2015. Crowdsourced data management industry and academic perspectives. *Foundations and Trends in Databases* 6(1-2):1–161.
- Martin, D.; Hanrahan, B. V.; O’Neill, J.; and Gupta, N. 2014. Being a turker. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, 224–235. ACM.
- McInnis, B.; Cosley, D.; Nam, C.; and Leshed, G. 2016. Taking a hit: Designing around rejection, mistrust, risk, and workers’ experiences in amazon mechanical turk. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2271–2282. ACM.
- Metaxa-Kakavouli, D.; Rusak, G.; Teevan, J.; and Bernstein, M. S. 2016. The web is flat: The inflation of uncommon experiences online. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2893–2899. ACM.
- Mitra, T.; Hutto, C.; and Gilbert, E. 2015. Comparing person-and process-centric strategies for obtaining quality data on amazon mechanical turk. In *Proceedings of the 33rd*

Annual ACM Conference on Human Factors in Computing Systems, 1345–1354. ACM.

Nielsen, J. 2000. Why you only need to test with 5 users.

Preece, J.; Rogers, Y.; Sharp, H.; Benyon, D.; Holland, S.; and Carey, T. 1994. *Human-computer interaction*. Addison-Wesley Longman Ltd.

Ross, L. 1977. The intuitive psychologist and his shortcomings: Distortions in the attribution process. *Advances in experimental social psychology* 10:173–220.

Rzeszotarski, J. M., and Kittur, A. 2011. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 13–22. ACM.

Sharp, H.; Jenny, P.; and Rogers, Y. 2007. Interaction design:: beyond human-computer interaction.

Sheng, V. S.; Provost, F.; and Ipeirotis, P. G. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 614–622. ACM.

Stephane, K., and Morgan, J. 2015. Hire fast & build things: How to recruit and manage a top-notch team of distributed engineers. In *Upwork*.

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

Wilson, J. Q., and Kelling, G. L. 1982. Broken windows. *Critical issues in policing: Contemporary readings* 395–407.

Wu, S.-Y.; Thawonmas, R.; and Chen, K.-T. 2011. Video summarization via crowdsourcing. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, 1531–1536. ACM.

Appendix A: Tasks

Three tasks the requesters designed

- *Marijuana legalization (classification)* (Metaxa-Kakavouli et al. 2016): Requesters designed a task to determine ten websites’ stances on marijuana legalization (Pro-legalization, Anti-legalization, or Neither). We chose this task because it captures a common categorical judgment task with forced-choice selection: other examples include image labeling (Deng et al. 2009) and sentiment classification (Callison-Burch and Dredze 2010). Similar to many such tasks, this one includes several edge cases, for example whether an online newspaper that neutrally reports on a pro-legalization court decision is expressing a ‘Pro-legalization’ or ‘Neither’ stance. Marijuana Legalization task included 10 URLs.
- *Query answering (search)* (Bernstein et al. 2012): Requesters were given a URL with a set of search queries that led to the URL and designed a task asking workers to copy-paste text from each webpage (URL) that answers those queries. One such query was “dog body temperature” (correct response: 102.5°F). We chose this task to be representative of search and data extraction tasks on Mechanical Turk (Cheng, Teevan, and Bernstein 2015). The challenge in designing this task correctly is in providing careful instructions about how much information to extract. For instance, without careful instruction, workers would often conservatively capture a large block of text instead of just the information needed (Bernstein et al. 2012). Query answering (search) included 8 search task URLs.
- *Video summarization (description)* (Kim and Monroy-Hernandez 2016; Wu, Thawonmas, and Chen 2011; Lasecki, Kushalnagar, and Bigham 2014): Requesters were given a YouTube video and requested to design a task for workers to write a text summary of the video. This was the most open-ended of the three tasks, requiring free-text content generation by the workers. To achieve a high-quality result, the task design needed to specify desired length, style, and example high and low quality summaries of a complex concept Video Summarization included 1 URL.

Appendix B: Figures

Figure 1: Workers provided feedback on tasks that ranged from clarity to payment and user interface

Category	Worker’s Feedback
Unclear or vague instructions	“This is really vague on what to do for summary, doesn’t say what format you want, how long it should be.” “The linked page has the question what does organic mean repeated several times, but it never really answers the question or explain what organic means.” “I had to skip one that had no possible content to answer a question about nutrition. This should be addressed in the instructions, as well as what do if the URL is broken etc.”
Missing examples	“Generic Example could help with general formatting and guidelines. More detailed instructions.” “The task is simple, and easy to follow. One suggestion would be to have examples of what they are looking for.”
Incorrect spelling and grammar	“Instructions are clear enough, but could use better grammar. The last sentence should be restated as the beginning does not make sense.” “Grammar and spelling errors are a big turnoff”
Poor or broken interface design	“Clickable links, and the paper isn’t exactly on topic, which makes the instructions problematic. I understand you can’t cover everything in the instructions, but the Pareto principle would say you can hit a lot of it if you put in a minimal amount of effort.” “Site’s buy section not open for India.”
Others	Payment: “40min for 0.03\$?” Praise: “Easy to comprehend instructions, easy to do overall. This task states it’s purpose well.”

Figure 2: With *prototype tasks*, all tasks are first launched to a small sample of workers to gather free-text feedback and example results. Requesters use prototype tasks to iterate on their tasks before launching to the larger marketplace.

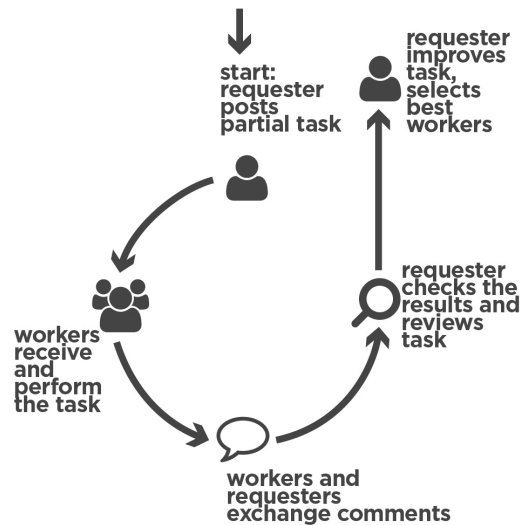


Figure 3: Daemo’s prototype task feedback from workers (right) can be used to revise the task interface (left). Requesters can review both the written feedback and the submitted sample results.

Drag components from the right sidebar and drop them here

Given a website, please give the website's stance on marijuana legalization

Website

<http://www.newsday.com/news/breaking/ny-medical-marijuana-bill-full-draft-1.8509071>

Which option below best matches the website's stance on marijuana legalization?

Pro-legalization

Anti-legalization

Neither

LIBRARY PROPERTIES FEEDBACK (14)

reworker1.reworker1
There are 3 articles on this website. It is not make clear what to do in the case of multiple articles. Because the instructions are not clear, I summarized the first one.
Posted 7 day(s) ago

npworker.6
The instructions are clear but it does not state what to do if much of the content is behind a paywall or a mandatory sign-up as is the case with this article.
Posted 1 day(s) ago

reworker1.reworker1
This is extremaly lengthy. There should be more specific instructions for what to do in the case of a legal bill. There should also be a timer and counter for the task itself.
Posted 7 day(s) ago

Figure 4: Prototype tasks appear alongside other tasks in workers’ task feeds on the Daemo crowdsourcing platform. Prototype tasks are paid tasks in which 3–7 workers attempt a task that will soon go live on the platform and give feedback to the requester to inform an iterative design cycle.

My Tasks Task Feed Forum Worker

Task Feed

\$3.00 Bitcoin YouTube Video Summarization Santosh 000
9 available tasks

\$5.00 **Prototype Task** Marijuana Legalization Stance Hu Yang 000
1 available task

\$2.00 **Prototype Task** Search Query John Doe 000
7 available tasks

\$1.00 SmartPhone Feature Review Jane 000
1 available task