The Need for Standardization in Crowdsourcing

Panagiotis G. Ipeirotis
New York University
panos@stern.nyu.edu

John J. Horton
Harvard University
oDesk Corporation
john.joseph.horton@gmail.com

ABSTRACT

Crowdsourcing has shown itself to be well-suited for the accomplishment of certain kinds of small tasks, yet many crowdsourceable tasks still require extensive structuring and managerial effort before using a crowd is feasible. We argue that this overhead could be substantially reduced via standardization. In the same way that task standardization enabled the mass production of physical goods, standardization of basic "building block" tasks would make crowdsourcing more scalable. Standardization would make it easier to set prices, spread best practices, build meaningful reputation systems and track quality. All of this would increase the demand for paid crowdsourcing—a development we argue is positive on both efficiency and welfare grounds. Standardization would also allow more complex processes to be built out of simpler tasks while still being able to predict quality, cost and time to completion. Realizing this vision will require interdisciplinary research effort as well as buy-in from online labor platforms.

Bios

Panos Ipeirotis has been using Mechanical Turk for his research (mainly to gather relevance judgments) since 2006, and over time the platform transformed from a research tool into a topic for research. He is interested in quality management and scalability issues related to crowdsourcing.

John Horton uses economics to explore the growing phenomena of people working online. He is particularly interested in the use of online labor markets as tools for economic development.

INTRODUCTION

The academic community and a growing number of firms are looking to paid crowdsourcing to solve problems. The problems being solved vary, but what they all have in common is one or more sub-problems that cannot be fully automated, and require human labor. This labor demand is being met by workers recruited from online labor markets such as Amazon Mechanical Turk, Microtask, oDesk and Elance or from casual participants recruited by intermediaries like CrowdFlower and CloudCrowd. In these markets, buyers and sellers have great flexibility in the tasks they propose and the making and accepting of offers.

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

The flexibility of online labor markets is similar to the flexibility of traditional labor markets. In both markets, buyers and sellers are free to trade almost any kind of labor at almost any terms. However, an important distinction between online and offline is that once a worker is hired off an offline, traditional market, they are *not* allocated to tasks via a spot market. Workers within firms are employees who have been screened, trained for their jobs and are have incentives for good performance—at a minimum, poor performance can cause them to lose their jobs. Furthermore, for many jobs—particularly those focusing on the production of physical goods—good performance is very well defined, in that workers must adhere to a standard set of instructions. This standardization of tasks is the essential feature of modern production and how it can be applied to crowdsourcing is the focus of our paper.

With task standardization, innovators like Henry Ford could ensure that hired workers—after suitable training—could complete those tasks easily, predictably and in a way that workers can be easily replaced with others, similarly trained. To return to paid crowdsourcing, most of the high demand crowdsourcing tasks are low-skilled and require workers to closely and consistently adhere to instructions for a particular, standardized task. As it currently stands, existing crowdsourcing platforms bear little resemblance to Henry Ford's car plants. In crowdsourcing markets, the factory would be more like an open bazaar where workers could come and go as they pleased, receiving or making offers on tasks that different in their difficulty and skill requirements ("install engines!", "add windshields!", "design a new chassis!") for different rates of pay—and with different pricing structures (fixed payment, hourly wages, incentives etc.). Some buyers would be offering work on buses, some on cars, some on lawnmowers. Reputations would be weak and easily subverted. Among both buyers and sellers, one can find scammers; some buyers are simply recruiting accomplices for nefarious activities.

The upside of such a disorganized market is that workers and buyers have lots of flexibility. There are good reasons for not wanting to just recreate the on-line equivalent of single-firm factory. However, we do not think it is an "either-or" proposition. In this paper, we discuss ways that we can have more structure on a marketplace platform, without undermining its key advantages. In particular, we believe that greater task standardization, a cultivated garden approach to work-pools and a market-making type work allocation mechanism to help arrive at prices could help us build scalable human-powered systems that meet real-world needs.

The rest of the paper is structured as follows. First, we give a

brief overview of the current status of on-line labor marketplaces. Next, we argue that the standardization of simple tasks can lead to immediate benefits in terms of pricing, speed, and easy adoption of best practices. Then, we discuss what are the benefits of constructing advanced tasks using workflows of standardized tasks, and discuss the role of the marketplace owners in properly designing the market for optimal outcomes for all participants. Finally, we conclude by presenting our views on the nature of platforms and other third parties (including researchers) and sketch out areas for future research.

CURRENT STATUS

In addition to the academic interest in paid crowdsourcing, an expanding ecosystem of firms use crowdsourcing for some business task, offer themselves as a platform for labor or help intermediate between labor pool and would-be users of crowdsourcing. A crowdsourcing industry group, Crowdsortium was recently formed and some of the larger players recently organized a conference, CrowdConf that was well-attended, attracting start-ups, academics and investors.

Despite the excitement and apparent industry maturation, there has been relatively little innovation—at least at the micro-work level—in the technology of how workers are allocated tasks, how reputation is managed and how tasks are presented etc. As innovative as MTurk is, it is basically unchanged since its launch. The criticism of MTurk—the difficulty of pricing work, the difficulty in predicting completion times and gaining quality, the inadequacy of the way that workers can search for tasks—are recurrent and still unanswered. Would-be users of crowdsourcing often fumble, with even technically savvy users getting mixed results. Best practices feel more like folk wisdom than an emerging consensus. Even more troubling, there is some evidence that at least some markets are becoming inundated with spammers.

One part of the crowdsourcing ecosystem that appears to be thriving is the "curated garden" approach used by companies like uTest (testing software), MicroTask (quality assurance for data entry), CloudCrowd (proofreading and translation), and LiveOps (call centers). These firms recruit and train workers for their standardized tasks and they set prices of both sides of the market. Because the task is relatively narrow, it is easier to build meaningful, informative feedback and verify ex ante that workers can do the task, rather than try to screen bad work out ex post. While this kind of control is not free, practitioners gain the scalability and cost savings of crowdsourcing without the confusion of the open market. The downside of these curated pools is that access as both a buyer and seller is limited. One of the great virtues of more market like platforms is that they are democratic and easy to experiment on. The natural question is whether it is possible to create labor pools that look more like curated gardens—with well defined, standardized tasks—and yet are still relatively open, both to new buyers and sellers?

STANDARDIZING BASIC WORK UNITS

Currently, the labor markets operate in a completely uncoordinated manner. Every employer generates its own work request, prices the request independently, and evaluates the answers separately from everyone else. Although this approach have some intuitive appeal in terms of worker and employer flexibility, it is a fundamentally inefficient approach.

- Every employer has to implement from scratch the "best practices" for each type of work. For example, there are multiple UI's for labeling images, or for transcribing audio. The long-term employers learn from their mistakes and fix the design problems, while newcomers have to learn the lessons of bad design the hard way.
- Every employer needs to price its work unit without knowing the conditions of the market and this price cannot fluctuate without removing and reposting the tasks.
- Workers need to learn the intricacies of the interface for each separate employer.
- Workers need to adapt to the different quality requirements of each employer.

We believe that the efficiency of the market can increase tremendously if there is at least some basic standardization of the common types of (micro-)work that is being posted on online labor markets.

So, what are these common types of (micro-)work that we can standardize? Amazon Mechanical Turk lists a set of basic templates[1], which give a good idea of what tasks are good candidates to standardize first. The analysis of the Mechanical Turk marketplace [4] also indicates a set of tasks that are very frequent on Mechanical Turk and are also good candidates to standardize.

We can draw in parallel with engineering: In mechanics, we have a set of "simple machines"[2], such as screws, levers, wheel and axle, and so on. These simple machines are typically standardized and serve as components for larger, significantly more complicated creations. Analogously, in crowdsourcing, we can define a set of such simple tasks, standardize them, and then build, if necessary, more complicated tasks on top.

What are the advantages of standardizing the simple tasks, if we only need them as components?

First of all, as mentioned above, there is no need for requesters to think on how to create the user interfaces and best practices for such simple tasks. These standardized tasks can be, of course, revised over time to reflect our knowledge on how to best accomplish them.

Second, and potentially more important, these simple tasks can be traded in the market in the same way that stocks and commodities are currently traded in financial markets. In stock markets, the buyer does not need to know who is the seller, or whether the order was fulfilled by a single seller or multiple ones: it is the task of the market maker to match and fulfill buy and sell orders. In the same way, we can have a queue of standardized tasks that need to be completed, and workers can complete them at any time, without having to think about the reputation of the requester or to (re-)familiarize themselves with the task. This should lead to much more efficient task execution.

A third advantage of standardized work units is that pricing becomes significantly simpler. Instead of "testing the market" to

[1] https://requester.mturk.com/bulk/hit_templates
[2] http://en.wikipedia.org/wiki/Simple_machine

see what price points leads to an optimal setting, we can instead have a very "liquid" market with a large number of offered tasks and a large number of workers that work on these tasks. This can lead to a stock-market-like pricing. The tasks get completed by the workers, in priority order according to the offered price for the work unit: the highest paying units get completed first. So, if requesters want to prioritize their own tasks, they can simply price them higher than the current market price. This corresponds to an increase in demand, which moves up the market price. On the other hand, if no requesters post tasks then, once the tasks with the highest prices get completed, then we automatically move to the tasks that have lower price associated with them. This corresponds to the case where the supply of work is higher than the demand, and market prices for the work unit move down.

In cases where there is not enough "liquidity" in the market (i.e., when the workers are not willing to work for the posted prices), then we can employ automated market makers [6], such as the ones currently used by prediction markets. The process would then operate like this: The workers identify the price for which they are willing to work. Then, the automated market maker takes into consideration the "ask" (the worker quote) and the "bid" (the price of the task), and can perform the trade by "bridging" the difference. Essentially, such automated market makers provide a subsidy in order for the transactions to happen. We should note that a market owner can typically benefit even in scenarios, where they need to subsidize the market through an automated market maker: the fee from a transaction that happens can cover the necessary subsidy which is consumed by the automated market maker.

We believe that having basic, standardized work units with highly liquid, high-volume markets can serve as a catalyst for companies to adopt crowdsourcing. Standardization can strengthen the network effects, can provide the basis for better reputation systems, can facilitate pricing, and can lead to the easier development of more complicated tasks that comprise of an arbitrary combination of small work units. We describe how to leverage basic work units next.

CONSTRUCTING AND PRICING COMPOSITE TASKS

Once we have some basic work units in place, we can start generating tasks that consist of multiple such units, to generate tasks that cannot be achieved with just using basic units.

Again we can draw the analogs from mechanical engineering: the "simple machines" (screws, levers, wheel and axle, and so on) can then be assembled together to generate machines of arbitrary complexity. Similarly, in crowdsourcing we can use these *standardized* set of "simple work units" that can be later assembled to generate tasks of arbitrary complexity.

Quality Assurance: Assume that we have a basic work unit for a task such as comment moderation, that guarantees an accuracy of 80% or higher (e.g., by screening and testing continuously the workers that can complete these tasks). If we want to have a work unit that has higher quality guarantees, we can generate a composite unit that uses multiple, redundant work units and relies on, say, majority vote to generate a work unit with higher quality guarantees.

Pricing Workflows: There is already work available on how to create [5] and control the quality [2] of workflows in crowdsourced environments. We also have a set of design patterns for workflows in general.[3] If we have a crowdsourced workflow that consists of standardized work units, we can also accurately price the overall workflow. We do not even have to reinvent the wheel: there is a significant amount of work on pricing combinatorial contracts [1] in prediction markets.[4] A workflow can be expressed as a combinatorial expression of the underlying simple work units. Since we know the price of standard units, we can easily leverage work from prediction markets to price tasks of almost arbitrary complexity. The successful deployment of Predictalot[5] by Yahoo! during the 2010 soccer World Cup, with the extensive real-time pricing of complicated combinatorial contracts, gives us the confidence that such a pricing mechanism is also possible for online labor markets.

Timing and Optimizing Workflows: There is already significant amount of work in distributed computing on optimizing execution of task workflows in Mapreduce-like environments [3]. This research should be directly applicable in an environment where the basic computation is performed not by computers but by humans. Also, since the work units will be completed through easy-to-model waiting queues, we can easily leverage the work from queuing theory to estimate how long a task will remain within the system: by identifying the critical parts of execution we can also identify potential bottlenecks and increase the offered prices for only the work units that critically affect the completion time of the overall task.

ROLE OF PLATFORMS

One helpful way to think about the role and incentives of on-line labor platforms is to consider that they are analogous to a commerce-promoting government in a traditional labor market. Most platforms levy an ad valorem charge and thus they have an incentive to increase the size of the total wage bill. While there are many steps these markets can take, their efforts fall into two categories: (1) remedying externalities and (2) setting enforceable standards and rules, i.e., their "weights and measures" function.

Remedying Externalities

An externality is created whenever the costs and benefits from some activity are not solely internalized by the person choosing to engage in that activity. A negative example is pollution—the factory owner gets the goods, others get the smoke—while a positive example is yard beautification (the gardener works and buys the plants, others get to enjoy the scenery). Because the parties making the decision do not fully internalize the costs and benefits, activities producing negative externalities are (inefficiently) over-provided, and activities producing positive externalities are (inefficiently) under-provided. In such cases, "government" intervention can improve efficiency.

Negative examples are easy to find in on-line labor markets—fraud is one example. Not only is fraud unjust, it also makes

[3] http://www.workflowpatterns.com/patterns/

[4] An example of a combinatorial contract: "Obama will win the 2012 election *and* will win Ohio" or "Obama will win the 2012 election *given that* he will win Ohio"

[5] http://predictalot.yahoo.com/

everyone else more distrustful, lowering the volume and value of trade. Removing bad actors helps ameliorate the market-killing problem of information asymmetry, as uncertainty about the quality of some good or service is often just the probability that the other trading partner is a fraud.

A positive example is honest feedback after a trade. Giving feedback is costly to both buyers and sellers: It takes time and giving negative feedback invites retaliation or scares off future trading partners. In the negative case, the platform needs to fight fraud—not simply fraud directed at itself but fraud directed at others on the platform, which has a negative second-order effect on the platform creator. In the positive case, the firm can make offering feedback more attractive, by offering rewards, making in mandatory, making it easier, changing rules to prevent retaliation etc.

There are lots of options in both the positive and negative case—the important point is that platform creators recognize externalities and act to encourage positive externalities and eliminate the negative ones. Individual participants do not have the incentives (or even the ability) to fix the negative externalities for all other market participants. For example, no employer has the incentive to publish his own evaluation of the workers that work for his, as this is a signal earned after a significant cost for the employer. This is a case where the market owner can provide the appropriate incentives and designs for the necessary transparency.

Setting Enforceable Standards

Task standardization will probably require buy-in from on online labor markets and intermediaries. Setting cross-platform standards is likely to be a contentious process, as the introduction of standards gives different incentives to different firms, depending upon their business model and market share. However, at least within a particular platform and ignoring their competitors, there is powerful incentive to create standards as they raise the value of paid crowdsourcing and promote efficiency. For example, the market for SMS's took off in the US only when the big carriers agreed on a common interoperable standard for sending and receiving SMS's across carrier's networks.

In traditional markets, market-wide agreement about basic units of measure facilitate trade. In commodity markets, agreements about quality standards serve a similar role, in that buyers know what they are getting and sellers know what they are supposed to provide. (For example, electricity producers are required to produce electricity adhering to some minimum standards before being able to connect to the grid and sell to other parties.) It should be clear that having public standards make quality assurance easier for the platform: enforcing standards on standardized units of work can be done much easier than enforcing quality standards in a wide variety of adhoc tasks. With such standards, it easier to imagine platform owners more willingly taking the role of testing for and enforcing quality standards for the participants that provide labor.

If we define weights and measures more broadly to include verification of claims, the platform role becomes even wider. They can verify credentials, test scores, work and payment histories, reputation scores and every other piece of information that individuals cannot credibly report themselves. Individuals are also not able to credibly report the quality of their work, but at least with an objective standard, validating those claims is possible.[6]

CONCLUSION

As we laid out in the paper, we believe that the key research challenge will be standardizing tasks. As standardization gains ground and best practices emerge, research will shift towards building labor-appropriate reputation systems and auctions as well as more complex work flows. Much of this research can be done at a micro level through researcher-initiated experiments. However, some "macro" research about platform-wide policies and institutions that cannot be tested anywhere but at the platform level. For these kinds of questions, we will need to rely on observational data or on large experiments conducted in conjunction with platform creators. Examples of this kind of research include dispute resolution policies, policies related to search and matching and reputation systems.

As our knowledge increases and platforms and practices mature, we expect far more work to be outsourced to remote workers. On the whole, we think this is a positive development, particularly because paid crowdsourcing gives people in poor countries access to buyers in rich countries, enabling a kind of virtual migration. If this form of increased virtual labor mobility has effects similar to those of increased real labor mobility, then the emergence of online labor markets could be transformative, as the welfare gains from liberalizing restrictions on labor mobility are truly enormous. Because of these potential welfare impacts, we view research on paid crowdsourcing as not only intellectually interesting, but also a way to expand economic opportunities for people who have relatively few opportunities due the accidents of birth and national borders.

References

[1] Y. Chen, S. Goel, and D. Pennock. Pricing combinatorial markets for tournaments. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 305–314. ACM, 2008.

[2] P. Dai, . Mausam, and D. Weld. Decision-theoretic control of crowd-sourced workflows. 2010.

[3] A. F. Gates, O. Natkovich, S. Chopra, P. Kamath, S. M. Narayanamurthy, C. Olston, B. Reed, S. Srinivasan, and U. Srivastava. Building a high-level dataflow system on top of map-reduce: the pig experience. *Proc. VLDB Endow.*, 2:1414–1425, August 2009.

[4] P. G. Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS*, 17:16–21, December 2010.

[5] G. Little, L. Chilton, M. Goldman, and R. Miller. Turkit: Tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 29–30. ACM, 2009.

[6] A. Othman, T. Sandholm, D. Pennock, and D. Reeves. A practical liquidity-sensitive automated market maker. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 377–386. ACM, 2010.

[6] For example, one of the main innovations made by oDesk was that they logged a worker's time spent on a task, enabling truthful hourly billing.