TurkBench: Rendering the Market for Turkers

Benjamin V. Hanrahan, Jutta K. Willamowski, Saiganesh Swaminathan, David B. Martin
Xerox Research Centre Europe
6 chemin de Maupertuis 38240 Meylan, France
Contact: ben.hanrahan@xerox.com

ABSTRACT
Crowdsourcing is a relatively new model of labor where both the workers and work providers are experiencing its growing pains. A dominant platform that implements this model of labor is Amazon Mechanical Turk (AMT). While AMT has evolved over the years, the changes have focused mainly on work providers and have not addressed the problems workers face (e.g. dealing with market volatility and unpaid time searching for work). In this paper we present *TurkBench*, a tool meant to provide workers with personalized market visualization and session management. We discuss the design philosophy of the tool, briefly discuss four Turkers' reaction to a demo, and outline future work.

Author Keywords
Crowdsourcing; Amazon Mechanical Turk

ACM Classification Keywords
H.5.2 User Interfaces

INTRODUCTION
Crowdsourcing is a relatively new model of labor where both the workers and work providers are experiencing its growing pains. Amazon Mechanical Turk (AMT) is one of the first and most dominant platforms for this model of labor. AMT has experienced a number of difficulties, for which Amazon has implemented several features to mitigate. However, Amazon has primarily focused on the problems of requesters (work providers), e.g. improving the Human Intelligence Task (HIT) creation interface and providing an Application Programming Interface (API) for posting and managing HITs. While, in our assessment, many of the issues experienced by the Turkers (workers) have not been addressed.

In fact, AMT provides a salient example of what can happen when the developers of a platform are also one of its users, especially when there is a power differential between this role and that of the other users. In the case of AMT, Amazon is both the platform developer and a major requester. Perhaps unsurprisingly they seem to have developed functionality primarily for requesters, the role they most easily identify with. Whether intentionally or not, this cements inequalities in the system. The effect runs deep and is even seen in the available API calls, where there is no support for Turkers.

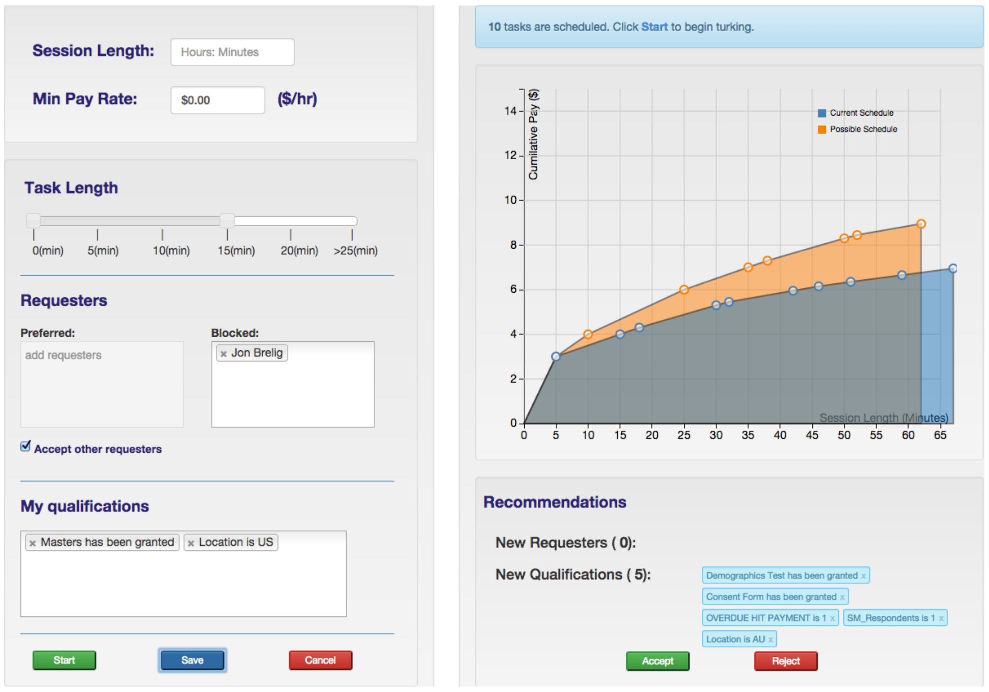
This phenomenon has several direct consequences for Turkers, many of which are rooted in the way the market is *rendered* to the Turkers. AMT does not display or prioritize HITs in a way that reflects the needs or wants of Turkers. This results in a large amount of unpaid work in finding HITs; difficulty in maintaining awareness in the face of market volatility; and a difficult on-boarding process [10]. This means that Turkers have to make compromises, e.g. choosing to do low-paying HITs instead of spending and potentially wasting time trying to locate higher paying ones.

These interface issues lead to larger consequences for AMT as a 'free market'[1] of labor, as the worker does not have sufficient information to make reasonably informed decisions. Lack of functionality and information also make many decisions more difficult, such as which employers are trustworthy, which jobs pay a reasonable wage, or which skills would be more profitable to acquire. Furthermore, Turkers do not have the means to declare these choices within the AMT platform, where requesters are provided mechanisms like blocking specific Turkers (for which there are broader consequences for Turkers). Turkers are put at a disadvantage (especially in comparison with requesters) through the structuring of information and functionality. When trying to frame AMT as a 'free market' of labor, one must remember that this labor market is in fact influenced a great deal by the functionality and information made available (or not) to the different roles. Not to mention the rules, procedures, and policies that surround a system like AMT.

In order to mitigate problems encountered by Turkers we built *TurkBench*. The goal of *TurkBench* is to reduce the difficulty in navigating the market by locating the most lucrative HITs, requesters, and qualifications available in the market. *TurkBench* also seeks to eliminate much of the unpaid search time by guiding users through an adaptive schedule. *TurkBench* provides several novel capabilities around adaptive scheduling and discovering new opportunities. We describe the related work and design of *TurkBench* in more detail in the following sections.

[1]We mean 'free market' in the mundane sense, where prices for labor and tasks are set by supply and demand.

CHI 2015, April 18 - 23 2015, Seoul, Republic of Korea
Copyright 2015 ACM 978-1-4503-3145-6/15/04$15.00
http://dx.doi.org/10.1145/2702123.2702279



10 tasks are scheduled. Click Start to begin turking.

Figure 1. Overview of Personalized Market Visualization interface.

RELATED WORK

Several tools have been developed to address the difficulties that requesters experience on crowdsourcing platforms. For example, *CrowdWeaver* [7] helps requesters create workflows to manage complex work. Similarly, Kulkarni et al. [9] devised *Turkomatic* to recruit workers in order to help requesters plan and solve complex jobs.

In contrast, there are only a few tools available to Turkers. The AMT platform itself suffers from usability problems and provides poor support for finding tasks [6]. As a result there is evidence that requesters take advantage of this and manipulate the search results to get HITs completed quickly [3].

There have been several studies that point to the imbalances in available information, power, and systems of redress [10, 11, 1, 13]. Researchers have also created a workers bill of rights[2] and posed the question, *"Can we foresee a future crowd workplace in which we would want our children to participate?"* [8]. Others have simply asked that tools frame their workers as more than just faceless computation [11, 12].

Of the tools that have been developed for Turkers, *TurkOpticon* is foremost. *Turkopticon* [5] helps workers identify "fair" requesters through collaborative ratings. Ipeirotis [4] has also done extensive work in defining the AMT market, and produced a site that Turkers use to gauge aspects of AMT[3]. Along the lines of *TurkBench*, Callison-Burch created *Crowd-Workers* in order to provide more transparency to the hourly wage of HITs [2]. In addition to the tools developed by researchers, Turkers have created several scripts and plugins to make turking easier. For example, *MTurkList*[4], which helps Turkers sort tasks based on pay; and *TurkAlert*[5] which notifies its users when certain HITs become available. In fact, the creating of scripts and plugins is a feature of the Turker culture, with entire sites dedicated to sharing them[6].

Similar to these tools and scripts, our goal is to facilitate turking. Also like these tools we must operate within the constraints of the AMT infrastructure, which provides limited support for Turkers. Tools have dealt with this problem in a variety of ways, *Turkopticon* collects user ratings in order to communicate requester reliability, adding the information directly to HITs. While, *Crowd-Workers* tracks the hourly rate of HITs by instrumenting the browser to record how long HITs take. Still others (e.g. *MTurkList* and *TurkAlert*) crawl the available HITs to provide ongoing information about the status of the market. We combine several of these methods in that we crawl AMT, record usage statistics, and augment the AMT interface. The various compromises that these tools must make due to the limitations of the AMT infrastructure, have an impact on their potential effectiveness and accuracy.

TURKBENCH DESIGN

TurkBench provides Turkers with a personalized *rendering* of AMT (e.g. available HITs, requesters, and qualifications), aiming to minimize unpaid work and increase earnings. More concretely, *TurkBench* is designed to eliminate search time between HITs, reduce the impact of market volatility, create an on-boarding path, and locate the highest paying HITs. While there are certainly more aspects to fulfilling work (e.g.

[2]`http://turkwork.differenceengines.com`
[3]`http://mturk-tracker.com/`

[4]`http://www.mturklist.com/`
[5]`http://www.turkalert.com/`
[6]`http://mturkscripts.com/`

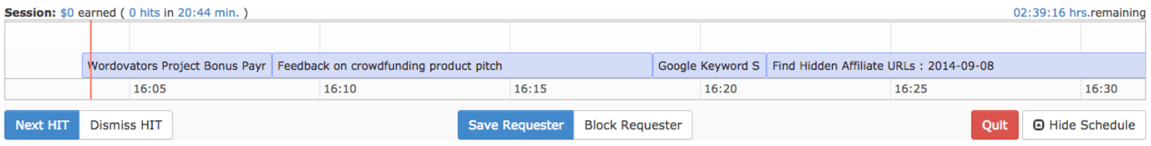

Session: $0 earned (0 hits in 20:44 min.)
02:39:16 hrs.remaining
Wordovators Project Bonus Payr
Feedback on crowdfunding product pitch
Google Keyword S
Find Hidden Affiliate URLs : 2014-09-08
16:05
16:10
16:15
16:20
16:25
16:30
Next HIT
Dismiss HIT
Save Requester
Block Requester
Quit
Hide Schedule

Figure 2. An example session displayed in the Session Manager.

education, entertainment, acquiring desirable skills), we feel that the two largest problems Turkers experience in AMT are the very low wages and the large amount of unpaid, invisible work involved. It is our hope that through using our tool, or similar ones, Turkers will gain the ability to find reasonably paying HITs and avoid exploitative requesters.

TurkBench is comprised of three primary components: the *Personalized Market Visualization*, where Turkers specify their settings and view the current state of the market; the *Session Manager*, which guides Turkers through automatically constructed work sessions; and the *Scheduler*, which crawls and constructs adaptive schedules for Turkers, based both on the volatile market of HITs and Turkers' settings.

Personalized Market Visualization

The *Personalized Market Visualization* (Figure 1) makes it easier to maintain awareness of AMT's volatile market and also provides support for on-boarding. It is implemented as a standalone webpage separate from the AMT interface.

The personalized *rendering* of AMT is done by collecting several settings from the Turker and displaying two possible schedules. Settings are located on the left of the interface and include the desired session length, minimum acceptable pay rate, preferred task length, preferred/blocked requesters, and any qualifications that the Turker has. Once these are specified, *TurkBench* constructs and displays the Turker's projected schedule (shown in grey) and an optimal schedule regardless of settings (shown in orange). This visualization helps the Turker to determine whether Turking is 'worth it' right now by displaying their projected pay rate and its relation to the optimum. If the optimum pay rate is higher than the projected pay rate, then *TurkBench* makes recommendations of 'missed' requesters and/or qualifications under the graph, which also helps to create an on-boarding path. The graph is interactive, in that Turkers are able to get specific information about each HIT by hovering over its respective data point. The combination of these features also enables Turkers to explore how their settings impact their pay. For example, Turkers are able to perform *what-if* analyses to gauge the value of different qualifications (or any of their settings). This assists the Turker to personalize the *rendering* of the AMT market based on their own priorities. Once the Turker is satisfied with their projected schedule, they can begin turking by clicking *Start*, which constructs their actual schedule and brings them to the *Session Manager*.

In regards to how we are calculating pay rate, we estimate it with increasing accuracy as Turkers use the system. In the worst case we use the *Time Allotted* as an estimate for the time required, however, once Turkers complete tasks with the *Session Manager* plugin installed, we are able to improve this estimate. We do this in two ways: 1) if the Turker *has not* completed any of the specific HITs then we use the global average of all Turkers; 2) however, if the Turker *has* completed at least one of the specific HITs then we use their individual average. We acknowledge that this will result in some false negatives, in that some HITs that have lots of *Time Allotted* will be unfairly rated as low paying tasks. However, in our experience our tool constructs realistic schedules that locate relatively high paying HITs. If we find that too many HITs are being unfairly rated we will introduce an estimation mechanism that Turkers can participate in.

Session Manager

The *Session Manager* (Figure 2) guides Turkers through their schedule of HITs, thereby eliminating unpaid search time between HITs. The personalized schedule is adaptive, in that it adjusts to market fluctuations in terms of available HITs. This feature relieves the effect of market volatility while working.

The *Session Manager* displays a timeline[7] which contains one cell for each scheduled HIT. The timeline indicates the current time with a red line, giving Turkers an idea at how they are performing in regards to the estimates. Turkers can advance to the next scheduled HIT by either clicking *Next HIT* to mark the current HIT as completed, or *Dismiss HIT* to remove it from the schedule. If a Turker likes or dislikes the current HIT, they can either save or block the corresponding requester for future scheduling. At any point in time the Turker can quit the session, freeing up any HITs that have not been completed for other Turkers. The top of the toolbar contains aspects of the current session, i.e. earnings, number of HITs completed, and time elapsed/remaining.

The *Session Manager* also keeps track of the amount of time that each HIT takes to complete. This is done by tracking the time between when the Turker accepts the HIT and marks the HIT as complete. This length is what is used by our system to calculate the aforementioned pay rate of each HIT.

Task Scheduler

The engine behind the *Personalized Market Visualization* and *Session Manager* is our *Task Scheduler*. The *Task Scheduler* crawls all of the HITs on AMT periodically and ranks them according to their estimated pay rate. When constructing schedules it selects the earliest available Turker and then walks through the sorted list of HITS, testing whether each HIT satisfies the Turker's settings. As HITs fail to satisfy the settings, the reasons for failure are logged. This log is the basis for any recommendations given to the Turker, which aids in discovering new opportunities. The scheduler runs every few minutes and continually prunes or updates Turkers'

[7]https://github.com/almende/vis

schedules as tasks appear, are completed, or expire. This continually optimizes each Turking session, eliminating the need to maintain awareness of market volatility.

TURKER REACTION

Before deploying *TurkBench* we conducted four 30 minute demos with individual Turkers to get a first impression of the tool. We used their settings, preferences, and resulting schedules to achieve a realistic condition for the demo.

The most enthusiastic reactions came from Turkers 1 and 3. Turker 1 was a female that had been turking on and off for a year and Turker 3 was a male who had been turking for 3-4 months. Neither Turker used many customizations and turking was not their primary income. Both Turkers reported difficulty finding HITs and they were the only participants who abandoned turking sessions because they could not find HITs. Both Turkers spent a great deal of time searching for HITs and were excited to see the session manager walking through a schedule. During the demo *TurkBench* found a pay rate of $6 an hour for Turker 3 (his typical rate was $2-3 an hour) and he intended to work after our interview.

Turker 4 was a male who had been turking for 1 year, used a modest amount of scripts, and drew his primary income from turking. He set a target of $20 per day and did not always meet it. He was not as enthusiastic as the previous two Turkers since he had no trouble knowing which HITs to find. However, he did have trouble with the mechanics of quickly locating HITs and thought he would use *TurkBench* to help him gauge the difficultly and speed of hitting his quota, as well as help him quickly execute a series of HITs.

Turker 2 was a male who had been turking for 1 year, used extensive customizations, and did not draw a primary income from turking. Turker 2 was the most experienced of the Turkers. He maintained relationships with requesters that directly recruited him through email so that he would complete specific upcoming tasks. His target was to earn $20 a day, which he said he could normally do within 2-3 hours without much trouble. Turker 2 said that he would probably not use the session manager since he completed longer tasks outside of the AMT platform, but would use the market visualization to explore which qualifications would help him earn more.

Taking the impressions from our demo into consideration, we believe that the use of *TurkBench* will provide a better understanding of the market and result in higher pay for Turkers by reducing search time, especially for less experienced Turkers.

DISCUSSION

The goals of *TurkBench* are to eliminate unpaid work, make it easier to maintain awareness of market volatility, and support on-boarding/discovery. These goals are shaped by our analysis of Turkers' position within AMT. Through the design and argumentation that we have presented in this paper, we hope to shine further light on the impact of the design decisions made by the platform developer, particularly in regards to the balance of features across roles. We acknowledge the extreme difficulty in foreseeing the challenges of each user role when developing an interface for a system like AMT. However, when the user community signals that they have problems with their level of agency, we draw attention to the fact that the platform developer clearly has the fullest ability to grant more agency to each of the user roles.

Our near term goal is to deploy *TurkBench* to Turkers. Through this deployment we hope to answer several questions around the efficacy and resulting impact of our tool. The easier of these questions relate to individual effects, such as whether *TurkBench* decreases unpaid work activity. The more difficult questions are around broader impacts on the market. For example, will the more powerful, personalized *rendering* of the market reduce the number of low paying tasks?

REFERENCES

1. Bederson, B. B., and Quinn, A. J. Web workers unite! addressing challenges of online laborers. In *CHI 2011 EA*, ACM (2011), 97–106.
2. Callison-Burch, C. Crowd-workers: Aggregating information across turkers to help them find higher paying work. In *HCOMP-2014* (November 2014).
3. Chilton, L. B., Horton, J. J., Miller, R. C., and Azenkot, S. Task search in a human computation market. In *ACM SIGKDD Workshop on Human Computation*, ACM (2010), 1–9.
4. Ipeirotis, P. G. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads 17*, 2 (2010), 16–21.
5. Irani, L. C., and Silberman, M. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *CHI 2013*, ACM (2013), 611–620.
6. Khanna, S., Ratan, A., Davis, J., and Thies, W. Evaluating and improving the usability of mechanical turk for low-income workers in india. In *DEV '10*, ACM (2010).
7. Kittur, A., Khamkar, S., André, P., and Kraut, R. Crowdweaver: Visually managing complex crowd work. In *CSCW 2012*, ACM (2012), 1033–1036.
8. Kittur, A., Nickerson, J. V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M., and Horton, J. The future of crowd work. In *CSCW 2013*, ACM (2013), 1301–1318.
9. Kulkarni, A., Can, M., and Hartmann, B. Collaboratively crowdsourcing workflows with turkomatic. In *CSCW 2012*, ACM (2012), 1003–1012.
10. Martin, D., Hanrahan, B. V., O'Neill, J., and Gupta, N. Being a turker. In *CSCW 2014*, ACM (2014), 224–235.
11. ONeill, J., and Martin, D. Relationship-based business process crowdsourcing. In *INTERACT '13*. Springer Berlin Heidelberg, 2013, 429–446.
12. Quinn, A. J., and Bederson, B. B. Human computation: a survey and taxonomy of a growing field. In *CHI 2011*, ACM (2011), 1403–1412.
13. Silberman, M., Ross, J., Irani, L., and Tomlinson, B. Sellers' problems in human computation markets. In *SIGKDD Workshop on Human Computation*, ACM (2010), 18–21.