



Investigating the Amazon Mechanical Turk Market Through Tool Design

Benjamin V. Hanrahan¹, David Martin², Jutta Willamowski³ & John M. Carroll¹
¹*Pennsylvania State University, University Park, PA, USA (E-mail: bvh10@psu.edu);* ²*Xerox Research Centre Europe, Meylan, France;* ³*Naver Labs Europe, Meylan, France*

Abstract. We developed *TurkBench* to better understand the work of crowdworkers on the Amazon Mechanical Turk (AMT) marketplace. While we aimed to reduce the amount of invisible, unpaid work that these crowdworkers performed, we also probed the day-to-day practices of crowdworkers. Through this probe we encountered a number of previously unreported difficulties that are representative of the difficulties that crowdworkers face in both building their own tools and working on AMT. In this article, our contributions are insights into 1) a number of breakdowns that are occurring on AMT and 2) how the AMT platform is being appropriated in ways that, at the same time, mitigate some breakdowns while exacerbating others. The breakdowns that we specifically discuss in this paper, are the increasing velocity of the market (good HITs are grabbed within seconds), the high amount of flexibility that requesters can and do exercise in specifying their HITs, and the difficulty crowdworkers had in navigating the market due to the large amount of variation in how HITs were constructed by requesters. When the velocity of the market is combined with a poor search interface, a large amount in variation in how HITs are constructed, and little infrastructural support for workers, the resulting work environment can be frustrating and difficult to thrive in.

Keywords: Crowdsourcing, Amazon Mechanical Turk

1. Introduction

We designed and built *TurkBench* for crowdworkers on the Amazon Mechanical Turk (AMT) marketplace, the most dominant platforms for procuring labor from the crowd (Hanrahan et al. 2015). Crowdsourcing itself is a relatively new model of labor, where requesters post Human Intelligence Tasks (HITs) to be completed by a mostly anonymous crowd. Many aspects of how these labor markets work (or even how they *should* work) are still being discovered (Kittur et al. 2013; Felstiner 2011).

Our purpose in developing and deploying *TurkBench* was to investigate how the market is working and how it might function a bit better. The research question that we specifically report on in this work is: *What are the day-to-day practices of working on AMT and how does the system support or impede these practices?* As part of answering this question we undertook two activities: first, we wanted to reduce the amount of invisible, unpaid work that is a necessary part of crowdwork,

in part by addressing the imbalance in the amount of functionality provided to requesters and Turkers (previously reported in (Hanrahan et al. 2015)); second (and the topic of this article), we wanted to investigate the manner in which the AMT platform was impacting the practices and experiences of the crowdworkers.

As we conducted this research, we found that these two activities were closely intertwined. That is, initially we imagined that the imbalance in functionality would manifest itself primarily as missing information (Hanrahan et al. 2015), in that we hypothesized that crowdworkers simply did not have access to certain types of information. Based on our findings this is certainly true to an extent, however, Turkers (as crowdworkers refer to themselves on AMT) have found and developed methods to access much of the information that they need. Instead, many of their problems are centered around breakdowns in the platform/market structure and a lack of infrastructural support. So, we found that this imbalance is part of a larger phenomenon where requesters have greater access in terms of *information* (e.g. ratings), *functionality* (e.g. guided work flows and API access), and *organizational* levers (e.g. blocking Turkers).¹ These deficiencies have a large impact on the mechanics of Turkers' work practices and while Turkers go to great lengths as a community to resolve many of these difficulties, their efforts often serve to highlight their lack of support. We would like to note that while this imbalance is clear, it does not simply imply that requesters do not suffer from a lack of information and functionality, they do however start from a better place and are regularly receiving functional enhancements as AMT matures.

In this paper we present our evaluation of *TurkBench*, leveraging it as a technology probe (Hutchinson et al. 2003) into the experience of working on AMT. We have found that a technology probe like ours is especially useful when investigating a workplace that is highly distributed and digitally mediated. Through developing, deploying, and evaluating *TurkBench* we were able to construct a common ground with the distributed Turker community and gain better access to their practices. We leverage our experience with *TurkBench* to outline and illustrate developing breakdowns in the AMT marketplace.

In terms of the breakdowns that we found in the work of turking, we found that the velocity of the market is already quite fast and seems to be increasing due to various appropriations and tools (good HITs are grabbed within seconds). We also encountered the incredible flexibility that requesters can and do exercise in specifying their HITs and getting them completed. This flexibility added to the difficulty Turkers have in navigating the market due to the large amount of variation in how HITs were constructed by requesters. When the velocity of the market is combined with a poor search interface, a large amount in variation in how HITs are constructed, and little infrastructural support for workers, the resulting work environment can be frustrating and difficult to thrive in.

¹ Of course, these are not fully distinct categories as *information* and *organizational levers* often have their instantiation in *functionality*.

Investigating the Amazon Mechanical Turk...

The structure of this article is as follows. First we review the related work, we then explain the methodology of our evaluation. Next we briefly explain our design and approach, presenting the usage of the tool outlining the issues and practices that we discovered during the study. After we discuss our tool, we outline the breakdowns that we encountered over the course of developing and deploying *TurkBench*. Finally, we discuss the implications of our work.

2. Related work

There are a number of researchers trying to understand crowdworking through design research interventions. *TurkOpticon* (Irani and Silberman 2013) (a system for rating requesters and HITs) and We Are Dynamo (Salehi et al. 2015) (a site for community organizing and campaigns) are two prominent examples. Callison-Burch created the *Crowd-Workers* plugin in order to provide more transparency to the hourly wage of HITs (Callison-Burch 2014), a shared goal with *TurkBench*. However, our tool differentiates itself by aiming to facilitate how Turkers structure and find their work. Having a focus that is more inline with what Turkers are already doing, has led us to encounter many of the same issues that the Turker community itself faces.

Similarly to these interventions, we must operate within the constraints of the AMT infrastructure, which provides limited support for Turkers. These previous interventions have dealt with this problem in a variety of ways, *TurkOpticon* (Irani and Silberman 2013) collects user ratings in order to communicate requester reliability, adding the information directly to HITs. While, *Crowd-Workers* (Callison-Burch 2014) tracks the hourly rate of HITs by instrumenting the browser to record how long HITs take.

Previous research has also discussed the usability problems and poor support for Turkers on the AMT platform in terms of finding HITs (Khanna et al. 2010; Martin et al. 2014; Hanrahan et al. 2015). There is evidence that requesters take advantage of this poor support in at least one way, by manipulating the metadata of their unchosen HITs so that they are listed as new and appear at the top of the list of HITs (Chilton et al. 2010).

There have been several studies that point to the imbalances in available information, power, and systems of redress (Martin et al. 2014; O'Neill and David Martin 2013; Bederson and Quinn 2011; Silberman et al. 2010; Felstiner 2011). Researchers have also created a workers bill of rights,² which gathered comments from Turkers about their expectations for a more fair workplace. Much of these comments pertained to the conditions under which workers were working, for example, whether or not there should be a minimum wage and mechanisms for mediating the interactions between Turkers and requesters (e.g. providing

² <http://www.turkwork.differenceengines.com>

recourse for rejections). While our research deals with some of these same issues, they are mostly tangential to our investigation. We discuss more about the difficulties in using AMT in its current form, our findings are more to do with Turkers as users of the platform and less about what Amazon the company should provide workers in a workplace.

Researchers have also posed the question, “Can we foresee a future crowd workplace in which we would want our children to participate?” (Kittur et al. 2013), part of which focused on wages and career development. Others have simply asked that tools frame their workers as more than just faceless computation (O’Neill and Martin 2013; Quinn and Bederson 2011). Along these lines Salehi et al. (2015) built *Dynamo*, which seeks to help Turkers mobilize social campaigns around some of these and other issues. This effort helped to produce a Turker-led letter writing campaign that appealed to Jeff Bezos to improve Turker pay and rights, as well as signing up many leading academic requesters to an ethical code of conduct. In terms of trying to better understand the market, Ipeirotis (2010) has done extensive work in defining the AMT market, and produced a site that Turkers use to gauge aspects of AMT.³

TurkBench differs from these various efforts in both: the activities that it tries support, in that we are looking to directly support the implicit, collective locating of tasks and new opportunities quickly; and the various difficulties and breakdowns that we found in building and deploying *TurkBench*.

3. Method

We performed an investigation of Turkers’ work practices and experiences by using our prototype as a technical probe (Hutchinson et al. 2003). Probes, especially in HCI, have experienced a degree of ambiguity and flexibility (Gaver et al. 2004; Boehner et al. 2007), however, we feel that we are sufficiently in the bounds of what can be considered a technology probe. We chose to use a technology probe for our investigation for two reasons: first, we wanted to get experience in building tools for AMT in the same manner by which Turkers are doing so; and second, it is difficult to study and engage with Turkers’ work practices due to their distributed nature, we felt that having a technology that we were building for them would provide us with access and data about their practices.

In our evaluation we wanted to engage Turkers in their own work setting (Brown et al. 2011) and investigate their practices of working on AMT. We recruited 10 Turkers through forum posts on TurkerNation⁴ and by posting a recruitment HIT on AMT itself. We paid each of the participants \$10, our goal was to recruit 10 Turkers to use our tool, and we ultimately recruited 11 Turkers

³ <http://www.mturk-tracker.com/>

⁴ <http://turkernation.com/>

Investigating the Amazon Mechanical Turk...

(one Turker had to withdraw from the study due to computer problems). Three of our participants were recruited from the TurkerNation forum post and the remaining seven were recruited from the HIT posted to AMT. Each Turker participated in three activities: an introductory session, using the system, and an interview.

During the introductory session we: ensured that our system was correctly installed; demoed and explained the functionality; and made sure that the participants understood what we expected from them. We also used this opportunity to ask our participants about the current issues they face, how that matched with the assumptions of our tool, and what their current toolset was and for what reasons.

After the introductory session we asked that Turkers use the tool for at least 30 minutes during their work, meaning that they needed to locate and complete HITs using our tool. This broadly defined task meant that the Turkers were using *TurkBench* to implement their own practices and not what we imagined their work to be, in this way their own issues arose. We paid our participants as we were mindful that our tool might negatively impact their earning potential, since prototypes can have unforeseen problems and require a learning period.

The final activity, was to participate in an interview, via either email or a Skype interview. We did this because a portion of the Turkers expressed their discomfort with using Skype or the telephone, the main reason for concern cited by Turkers was one of privacy and discomfort. They also expressed skepticism about why we were doing our study and wondered what our motivations were. That said, five of the Turkers opted for the Skype interview. Regardless of the medium we began discussing whether or not they were able to use our tool successfully and why, and moved to discuss their more general experiences of working on and with AMT. These interviews lasted anywhere from 15 to 90 minutes over Skype or a single email to a thread of 4–5 emails.

Through this technology probe we were able to both evaluate our tool design and investigate the current practices of the Turkers within the group of participants.

4. TurkBench design and implementation

The aim of *TurkBench* is to reshape some of the imbalances between Turkers and requesters, mainly through addressing information imbalances to lower the opportunity cost of finding and executing work. Turkers often talk about the opportunity cost of finding new, higher paying tasks. This cost often causes them to take known, lower paying tasks in order to maximize their pay over time (Martin et al. 2014).

In large part lowering the opportunity cost requires automating the tedious tasks that Turkers engage in to locate this work. Aside from automation, we support two distinct activities exploring the AMT market and completing HITs. Exploration functionality makes it easier to determine the current market status and

discover new opportunities. Work session functionality walks Turkers through guided schedules, in order to eliminate search time by automatically finding new work.

TurkBench has three main components: the *Personalized Market Visualization*, where Turkers input their settings and explore the current labor market; the *Session Manager*, which guides Turkers through automatically constructed schedules; and the *Scheduler*, which constantly crawls AMT in the background and constructs schedules that adapt as HITs appear and disappear from the market. In the following sections we briefly discuss our implementation (for more detail see Hanrahan et al. 2015).

4.1. Personalized market visualization

The *Personalized Market Visualization* (Figures 1, 2, 3) serves several purposes: first, it gathers the requisite information to construct schedules for each Turker; second, it helps Turkers determine whether Turkering is ‘worth it’ right now; lastly, it helps Turkers to discover new opportunities. This portion of the tool is implemented as a separate web page from the AMT interface.

In order to construct meaningful, personalized schedules for Turkers, they must provide preferences (Figure 1): the length of the next session, their minimum

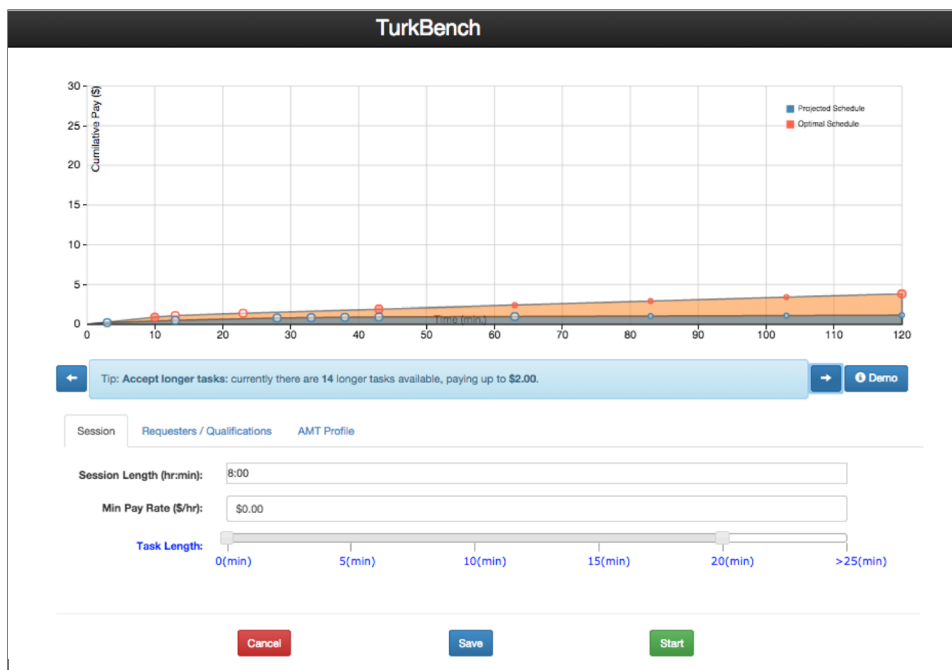


Figure 1. Overview of Personalized Market Visualization interface.

Investigating the Amazon Mechanical Turk...

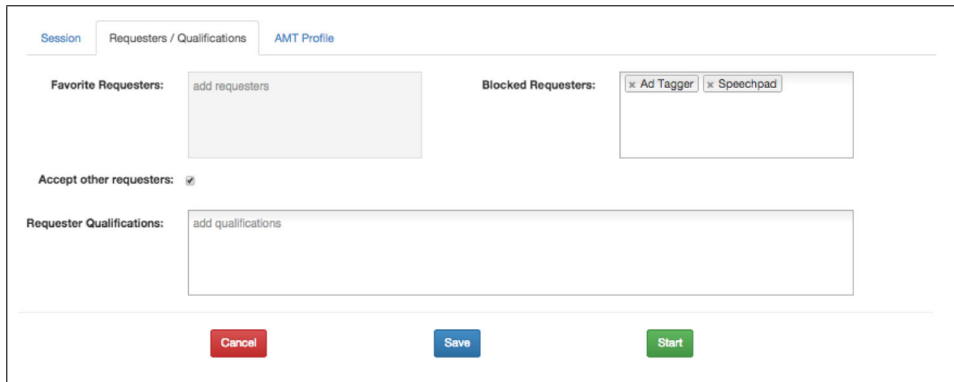
The image shows a web interface for managing requesters and qualifications. At the top, there are three tabs: 'Session', 'Requesters / Qualifications' (which is active), and 'AMT Profile'. Below the tabs, there are three main sections: 1. 'Favorite Requesters:' with a text input field containing 'add requesters'. 2. 'Blocked Requesters:' with a list of two items: 'x Ad Tagger' and 'x Speechpad'. 3. 'Accept other requesters:' with a checked checkbox. Below these is 'Requester Qualifications:' with a text input field containing 'add qualifications'. At the bottom, there are three buttons: 'Cancel' (red), 'Save' (blue), and 'Start' (green).

Figure 2. The information gathered on the requester tab, it contains the preferred/blocked requesters and the various requester qualifications they have obtained.

acceptable pay rate, their preferred HIT length. Turkers also provide information specific to requesters and their qualifications⁵) (Figure 2).

With this information we were able to construct ‘what-if’ schedules, which help Turkers determine if turking is ‘worth it’ right now (something that they typically discuss in forums (Martin et al. 2014)). These ‘what-if’ schedules are the: *projected schedule* which is highest paying schedule given the qualifications and preferences; and the *optimum schedule* which is the best case, regardless of the Turker’s preferences and qualifications.⁶ To support discovery of new opportunities indicated by the *optimum schedule*, we provide tips and information on aspects like potential qualifications or requesters prioritized by how much market potential each opportunity is worth.

4.2. Session manager

The *Session Manager* (Figure 4) is where the majority of work is done by the Turkers. This Google Chrome plugin sits on top of the AMT site and guides the Turker through an adaptive schedule of HITs. The plugin displays a timeline in which a single cell is representative of a single instance of a HIT and the size of the cell indicates its estimated length. Alongside the timeline there are several buttons to perform various operations on the HITs, notably:

Report HIT lets Turkers indicate that a HIT is no longer valid and should be removed from the scheduler.

⁵ Examples of qualifications include user based statistics like percentage of HITs accepted which is managed by Amazon, as well as the numerous qualifications assigned by requesters that range from the rather self explanatory, e.g. *Real Estate Link Qualification (NEW)*, to the incomprehensible, e.g. *Global_Test_NOTAUGR.CV.s*.

⁶ With the exception of aspects that the Turkers could not change, e.g. location

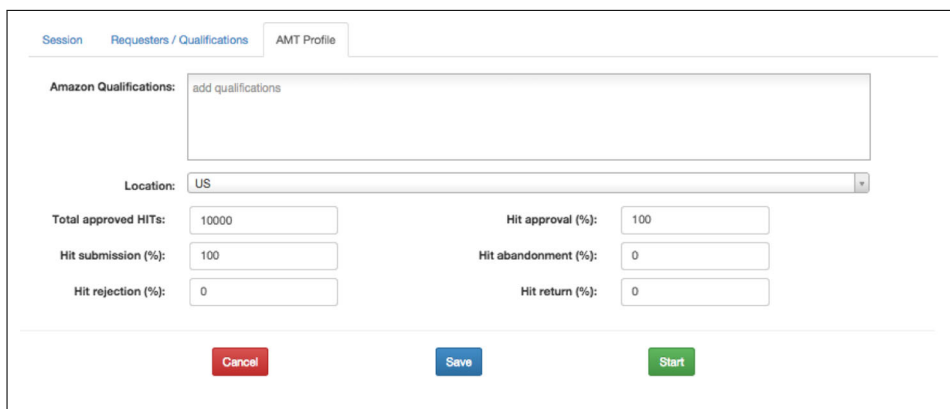


Figure 3. The information gathered on the Amazon tab, it contains the performance data and which Masters qualifications that the Turker has obtained.

Complete HIT is used by our system to gather when the HIT is completed, due to the variability in how the various HITs are completed on AMT (e.g. some HITs use custom websites) we found it necessary to make this an explicit user interaction.

The *Session Manager* also captures the amount of time that each Turker takes to complete each HIT. This is done by tracking the time between when the Turker clicks the *Goto HIT* button and when they click the *Complete HIT* button. These times inform how we estimate the time needed to complete a HIT for a specific Turker.

4.3. Scheduler

The primary component of the *Scheduler* is in fact the crawler that scrapes the available HITs. The biggest difficulty in building accurate schedules is avoiding our requests being throttled by AMT. To avoid being throttled achieve we continuously ran two crawls, each with a different purpose. The primary way that we discover new HITs is to crawl every 30 seconds on the list of HITs sorted by descending order of created date. Each run of this crawl terminates when we encounter a HIT that we have already seen and captured in our database. The second crawl is considerably slower, it does a complete scrape of every page of HITs

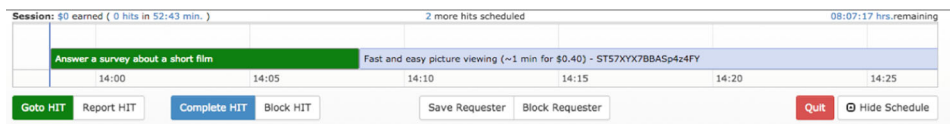


Figure 4. Overview of Session Manager, this is the part of the plugin that guides Turkers through a work session.

Investigating the Amazon Mechanical Turk...

every 30 minutes. The purpose of this crawl is primarily to expire any HITs that are no longer listed on AMT and take them out of consideration for the scheduler.

The scheduler itself uses the Turker's profile information to filter the available HITs for each Turker, it then ranks them according to estimated pay. As the Turker progresses through their schedule and new HITs become available, or previously scheduled HITs are exhausted, it continually adjusts and adds new HITs. Calculating the estimated pay rate is dependent on how we estimated time to complete a HIT.

There are three stages to how we estimate the time it takes to complete a HIT, each gets increasingly more accurate: 1) if no users of *TurkBench* have completed any of a type of HIT, then we use the metadata from the HIT itself (specifically the *Time Allotted*); 2) if the Turker has not completed any of the specific HITs then we use the global average of all Turkers; 3) however, if the Turker has completed at least one of the specific HITs then we use their individual average. This technique results in some false negatives, in that some HITs that have lots of Time Allotted will be unfairly rated as low paying HITs and will likely not be included in our schedules.

5. Breakdowns in the market

During our evaluation we found what we perceived to be several breakdowns in the AMT marketplace. We encountered some of these breakdowns due to some of the difficulties we had during our deployment and development. For instance, while four of our participants used *TurkBench* to successfully complete a number of HITs, others were unable to find HITs that they were accustomed to and did not complete any HITs. Sometimes this was due to conflicts with already heavily customized work environments, and in other cases due to the velocity of the market.

In terms of the usage of our tool, four of the ten Turkers used *TurkBench* for more than an hour, and two of these four Turkers used it for multiple days. Seven of the ten Turkers constructed a schedule and four of these successfully found and completed a total of 80 hits out of a possible 340. That said, *TurkBench* as a design probe helped us to identify practices and problems in the work practices of the Turkers and conditions of AMT as a workplace.

While previous literature is broadly aware that Turkers create scripts for each other (Martin et al. 2014), our participants made us aware of the extent that successfully operating in AMT marketplace depended on a number of tools that we were unaware of and also how they specifically made use of them. These tools are integral in finding and completing HITs in the relatively competitive AMT market. Some of the tools that our participants used were: *MTurkList*,⁷ helps Turkers

⁷ <http://www.mturklist.com/>

sort HITs based on pay; *TurkAlert*⁸ notifies its users when certain HITs become available; *Turkmaster*⁹ monitors and auto accepts HITs for specified requesters or URLs; *HIT Scraper*¹⁰ scrapes HITs and integrates with *TurkOpticon*. In fact, the creating of scripts and plugins is a feature of the Turker culture, with entire sites dedicated to sharing them.¹¹

The various compromises that all of these tools, including ours, must make due to the limitations of the AMT infrastructure, have an impact on their potential effectiveness and accuracy. Additionally, some of these tools compound the very problems that they were meant to solve (e.g. the velocity of the market). In fact, this reflects a general phenomenon, participants in AMT (requesters and Turkers both) stretch the limited functionality in creative and innovative ways to suit their needs but eventually hit the buffers, and the accumulative effects create their own problems (e.g. also the suite of plugins eventually cause browsers to crash). This is leading to breakdowns developing in the market.

We identified several key areas where there appear to be breakdowns in the AMT market and system. These breakdowns seem to be compounding, and the seams of the work that must be stitched together by the Turkers are growing wider and more difficult to deal with. In this section we discuss these areas, some of which *TurkBench* sought to alleviate and others that we found during the course of our evaluation.

5.1. Velocity of the market

One of the most prominent problems that Turkers are currently wrestling with is keeping up with the velocity of the market. The velocity of the market is certainly due in part to the number of Turkers that are actively working around the clock, and the small percentages of quality HITs that expert Turkers are willing to complete. However, the velocity of the market seems to have increased over time because Turkers themselves have built several tools to help them gain an advantage in the market and grab HITs more quickly.

A basic function of these tools and customizations is a way to keep abreast of any changes in the market, as well as claiming known HITs as fast as possible and in as great a number as possible. As a participant in our study outlined to us, it is quite possible that new Turkers would not even see the best HITs as it can take as little as 20 Turkers each claiming 25 HITs to exhaust a batch of 500 in seconds.

Things can be gone in 2-3 seconds $500/25 = 20$ people to fill up their accept queue - P3

⁸ <http://www.turkalert.com/>

⁹ <https://greasyfork.org/en/scripts/4771-turkmaster-mturk>

¹⁰ <https://greasyfork.org/en/scripts/2002-hit-scraper-with-export>

¹¹ <http://mturkscripts.com/>

Investigating the Amazon Mechanical Turk...

One good HIT that had 100 and they were gone in less than one second [...] you have to use plugins. - P4

Popular one-off HITs are grabbed as soon as they are posted to the market by whichever Turker's script is first to notice it, as one participant noted "good single HITs are taken in less than a second" - P5. This creates an additional barrier (and resulting gap) between expert and novice Turkers, meaning onboarding is all the more difficult.

There are two primary methods that these tools use to assist Turkers in their work. First, there are scrapers that Turkers have built that leverage their 'logged in' perspective. Meaning that they are using the fact that they are logged in locally to AMT and use this to filter HITs that they are qualified for. An example of this is the tool *HIT Scraper*. In addition, Turkers have various ways to claim HITs in their 'Accept Queue.' The 'Accept Queue' is how Turkers refer to the 25 HIT limit of accepted HITs, meaning that they are effectively reserved for the Turker until it expires. Some Turkers used simple claiming buttons, where once they found a HIT that they liked they hit a claim button and it used a URL convention to continuously grab HITs as they completed them. They also used tools like *Turkmaster* that would alert them when it found a HIT and could add it to their accept queue.

The effect of providing better support to deal with market velocity may seem to be exacerbating the problem as opposed to alleviating it, and one could argue that the velocity of the market will further increase as better support is provided. However, it may be that this velocity is a feature of these markets and that it only is emerging as a problem as more proper tool support is developed. In any case, as it stands right now, Turkers who have the tools to deal with this velocity have a distinct advantage over those who do not, which makes it difficult for new members to find (or even encounter) the most desirable work.

All that said, As one of our participants explained, there are many different ways that requesters create HITs that, wittingly or unwittingly, block the use of the accept queue and consequently make it more difficult to automate the accepting/claiming of HITs. For example, if there is a HIT that takes one minute to complete, but it expires five minutes after it is accepted, then a Turker cannot claim it as you would not be able to move through your queue fast enough. However, if there is a HIT that takes five minutes to complete and expires one hour after it is accepted, then you can claim this as you can move through your queue faster than they expire. This moves us to another developing breakdown, the flexibility with which requesters can create and design their HITs and the difficulty for Turkers in dealing with it.

5.2. Flexibility of HITs

Underlying these issues is the flexibility that requesters can and do exercise in specifying and creating HITs, which introduces additional difficulties and complexity for Turkers.

5.2.1. *Qualifications*

One example, is the requester appropriations of the qualification mechanisms for different, individual purposes. Turkers mentioned that requesters used qualifications for not only granting access to HITs, but also for blocking access to HITs, for example, qualifications with names such as ‘This is for worker XYZ all others will be rejected.’ We have also seen them as a mechanism through which to funnel a specific one-time payment.

Clearly, and justifiably, these qualifications are being appropriated by requesters for many uses, which is a good example of the creativity with which the limited functionality available to stakeholders is used to achieve multiple ends. The result of these large number of one-off customizations is that Turkers have an incredibly large number of qualifications that they must obtain, it was not uncommon for our participants to have earned over 400 qualifications. The end result is an overloading on qualifications, and the Turkers admitted that they didn’t remember, or in some cases know, what the majority of their own qualifications were for or even who issued them.

I don’t even know how many qualifications I have [...] requesters will just assign them for whatever reasons without communicating - P4

On top of this there are some HITs that lack appropriate qualifications to classify them as needing a specific skill. One example is that there are a number of highly paid HITs that require foreign language skills to successfully complete, however, there is no qualification assigned to the HIT that indicates this. This is reasonable from the requesters point of view, they avoid the overhead of administering qualifications and the HITs simply cannot be completed by a non-fluent speaker of the language. However, from the Turkers point of view it compounds the difficulty of navigating the market, since it degrades a primary mechanism for filtering HITs.

Another problem that Turkers face with qualifications is that it is difficult to determine which qualifications are currently available to obtain. For example, some qualifications are only open periodically, seemingly open when a requester needs to recruit more Turkers into their pool of workers. There is no easy way to find which qualifications require a test, are open, or a host of other factors.

It is difficult, almost impossible, to know which quals are available - P2

5.2.2. *HIT construction*

There is also a large variety in the ways that HITs are constructed. Requesters not only implement different strategies via the metadata for the HITs (e.g. setting the amount of time a Turker has to complete a HIT), Requesters can also choose where and how to host their HIT. That is, they can either use the AMT native tools to construct their HIT (and host it on AMT servers), or they can build a custom HIT hosted on an external website/server and require Turkers to navigate to the external site to complete the work. The majority of the time this is completely

Investigating the Amazon Mechanical Turk...

aboveboard, however, our participants said that there are times when it is done to nefarious ends, e.g. attempts at phishing that try to get personal information.

Due to this wide variety in how functionality was appropriated and HITs constructed, one strategy that Turkers developed in response was to specialize in a certain type of HIT. The Turkers that we talked with seemed to have a type of HIT that they preferred and tailored their environment for. Some clearly preferred large batch HITs and aimed their customizations at being able to quickly queue up what they were working on to claim the biggest chunk of the batch that they could. Others looked for things like surveys and tailored their environment more for discovery of new HITs.

5.3. Fragile ecosystem

The various tools that Turkers have constructed to counteract the breakdowns or imbalances in functionality that they encounter, has itself led to a developing breakdown. That is, Turkers rely on a fragile ecosystem of tools to do their work. The majority of these tools are ‘home brewed’ and live as plugins in the browser. Turkers cited that Chrome was their primary browser, however, they complained that it was prone to crash under the weight of the various plugins that they used. The Turkers that we talked to reported using upwards of 20 different plugins.

Turkers could not use all of these plugins at one time, as some of them interfered with each other. To mitigate these problems they utilized different profiles in Chrome for different stages in their work flow, e.g. a profile for locating survey HITs and different one for completing batches. One Turker went so far as to use multiple computers to manage the different contexts and stages of their work flow.

These conflicts were caused by one of two primary problems: first, due to multiple plugins modifying the same HTML page, there were many conflicting user interface elements that simply did not work together; second, and a much larger problem, was that multiple tools were relying on some kind of scraping mechanism. This led to Turkers experiencing the ‘Page Refresh Error,’ which is when AMT throttled their requests because multiple tools were making too many ‘page requests’ from Amazon’s servers. This practice of throttling when one user makes multiple, rapid HTTP requests is a standard reaction by websites. This was a large problem for the Turkers, and most of them had at most a cursory understanding of why they were running into problems and what the cause was, some did not know the reason at all. One participant said that they receive this problem within five minutes of starting a work session. To combat this the participant would constantly switch between activating all of their tools to gather HITs in their accept queue and then shut them off while they completed the queue. There were times when a particular Turker was unable to successfully complete survey HITs (even though they had completed the survey), because they were throttled and could not access the HIT, blocking them from entering the code and receiving payment.

There were also some pieces of information that Turkers did not have stable access to. One example is the *Returned Rate*, which is the number of HITs that a Turker returns after accepting. While this generally is not used by requesters, Turkers do need to be able to see this data as we have seen some requesters using it as a qualification. According to our participants, this specific metric was taken out of the Turker interface for reasons unknown to them. In order to regain access to the *Returned Rate* one Turker created a fake HIT that requires all of the metrics and therefore displays all of the metrics, various tools use the specific URL of this HIT to gather all of the metrics.

While these various techniques are quite resourceful given the limited set of functionality and infrastructure that is provided to Turkers, the various workarounds that each of them requires have an additive effect that, bit by bit, results in a fragile toolset that has become problematic for Turkers. However, these tools are also a core part of working as an expert on AMT and seem to be required to be able to make reasonable money on AMT. This tension, especially when coupled with how requesters are also stretching the functionality, indicates breakdowns in the AMT ecosystem.

6. Patching the breakdowns in AMT

These breakdowns lead to two main recommendations for better supporting the type of crowdwork that happens on AMT.

6.1. Infrastructural support

While there certainly are workarounds that can work on an individual level, there is not proper infrastructural support for tools like *TurkBench*, which try to serve the entire community and require information about HITs.

The velocity of a crowdsourcing market is inevitable as some tasks are simply much more desirable than others (pay rate being the most important dimension, but others also include ‘trustworthiness of the requester’, ‘attractiveness of the task’, etc.) and there are a limited number of these desirable tasks in relation to the supply of labor. Current tools are unable to keep pace with the velocity of the market without being throttled by AMT. This was cited as the ‘Number one issue’

You aren't allowed to hit an expert's request			
Requester: notings	HIT Expiration Date: Time Allowed:	Jul 3, 2015 (13 weeks 2 days) 52 weeks 1 day	Reward: HITs Available: 1
Sports Blind Dates 10			
Requester: notings	HIT Expiration Date: Time Allowed:	Apr 12, 2015 (1 week 4 days) 60 minutes	Reward: HITs Available: 1
How often do I hit			
Requester: notings	HIT Expiration Date: Time Allowed:	Apr 10, 2015 (1 week 2 days) 60 minutes	Reward: HITs Available: 1
Only for people with ratings			
Requester: notings	HIT Expiration Date: Time Allowed:	Apr 9, 2015 (1 week 1 day) 60 minutes	Reward: HITs Available: 1
Only for people with a high amount of hits			
Requester: notings	HIT Expiration Date: Time Allowed:	Apr 9, 2015 (1 week 1 day) 60 minutes	Reward: HITs Available: 1
Fake HIT			
Requester: notings	HIT Expiration Date: Time Allowed:	Apr 9, 2015 (1 week 1 day) 60 minutes	Reward: HITs Available: 1
Description: #gettask Keywords: none, none Qualifications Required: 0% HIT approval rate (%): 95 HIT abandonment rate (%): 95 Total approved HITs (%): 95 HIT return rate (%): 95 HIT submission rate (%): 95			

Some of the special HITs that plugin developers use to gather data.

Investigating the Amazon Mechanical Turk...

by Turkers with their workflow, this was especially true when they used plugins that gathered information to monitor HITs.

In fact, this was also the primary problem with *TurkBench*, and why our list of available HITs was not as up-to-date as Turkers needed. While we never impacted Turkers on their specific machine by design (we were specifically trying to avoid this), the scraper running on our server did get throttled if we tried to update too quickly. Based on the feedback of our participants, it seems the best hits are grabbed in less than 30 seconds, perhaps even less than 5 seconds and are missed by our crawler. From the workers perspective, features that help them more actively monitor the market status would be of great benefit to them (e.g. alerting when tasks they like become available), as this was a specifically cited capability that they used plugins to try to achieve.

We found that among our participants, Turkers would benefit from a structured, reliable stream of the currently available HITs and qualifications. This would likely eliminate the need for the majority of customizations and plugins used by Turkers, and would be useful to the subset of the Turker community that creates these plugins. Even relatively simple changes to the platform, e.g. providing live updates to saved searches or even notifying Turkers of new results to such a search would alleviate a large portion of their frustrations.

Amazon, and other maintainers of crowdsourcing platforms, should keep in mind that they are not simply providing a service to the requesters of HITs. Instead they are in the business of making connections between someone that needs something done, and someone that is willing to do it. Therefore the infrastructure should reflect this, being able to more easily and quickly locate HITs would be good for both Turkers and requesters more broadly, through better connecting ‘appropriate’ Turkers to requesters’ HITs.

6.2. Stakeholders customization

In a workplace like AMT, it seems that some amount of customization underpins the practice of how work gets done. That is in these environments and contexts, workers are doing quite different types of jobs that have quite different requirements from one another. The requirements for a worker that is filling out surveys, transcribing voice to text, or transferring receipt information are quite different. Workers that are doing short HITs require a more streamlined environment, where workers that are filling out surveys or doing long HITs for many different employers need help locating new opportunities. A specific example of this from AMT, is the limit of claiming 25 HITs at a time, for some workers 25 HITs can represent a few minutes worth of work, for others it might represent a few days’ worth. Meaning that different workers can potentially claim considerably larger blocks of work than others. Recognizing these different requirements, links very strongly to the need for proper infrastructural support so that workers are able to customize their work environment effectively.

7. Impact of the breakdowns in the market

As we have discussed, we feel that a number of appropriations and customizations that are happening on AMT are stretching the limited functionality provided to stakeholders and is uncovering, or exacerbating, breakdowns in the overall system. While developing and deploying *TurkBench* we saw evidence of the many ways that requesters creatively use qualifications and HIT design to get their specific work done by the crowd of Turkers. We experienced the difficulties that this can cause for Turkers first hand, the numerous one-offs accumulate quickly to make the system difficult to navigate. In this paper we have enumerated the difficulties Turkers experience and the workarounds that they employ.

One thing that struck us while encountering and analyzing these breakdowns is that all stakeholders in the workplace were acting and reacting logically, for example: it is difficult for good requesters to figure out how to best capitalize on the crowd's labor, so they appropriate the available functionalities in ways that best support their individual needs; there are not enough high paying, high quality HITs to go around, so the Turkers developed tools so that they could more quickly and easily find these high paying HITs; Amazon reacts to what appears to be malicious behavior towards their server by shutting down the offending party's access. However, when all of these various behaviors are viewed holistically, it shows the breakdowns in the market's ecosystem.

While, it seems that all stakeholders are experiencing some amount of difficulty or emergent breakdown in the ecosystem. We see that there is an asymmetry when it comes to Turkers, where Amazon and the requesters have greater access in terms of *information* (e.g. ratings), *functionality* (e.g. guided work flows and API access), and *organizational levers* (e.g. blocking Turkers). We would argue requesters do start with more capabilities and receive more attention from Amazon in terms of functional enhancements in response to the breakdowns that impact them.

This has the effect that, no matter which specialization a Turker takes or what customizations they make, a requirement of effective Turking was constant engagement and long hours on the platform. The Turkers that we talked to reported working 10–12 hour days and constantly checking for HITs. One participant set up alerts to wake them when their scripts found a certain HIT that they preferred (a similar behavior was reported by Gupta et al. 2014). That said, one Turker wanted it noted that their long hours are not necessarily bad because much of their time is spent idle waiting for quality HITs and they could do things like watch sports on television. However, this still means that they must, at some level, maintain engagement and availability to work. Traditionally, this 'on call' requirement is remunerated by an organization.

When designing systems for work that requires high levels of efficiency and throughput, designers need to keep in mind that there is a key difference between having access to information and being able to put that information to work. A concrete example of this can be seen in how Turkers find the available HITs

Investigating the Amazon Mechanical Turk...

(something very important to their work). The Turkers have in large part found a way to access this information through building tools that scraped AMT. So Turkers have *access* to this information, however, using these tools causes its own set of problems, i.e. they quickly get throttled by AMT which blocks them from effectively putting this information to work for optimizing their work flows. These deficiencies have a large impact on the mechanics of Turkers' work practices and while Turkers go to great lengths as a community to resolve many of these difficulties, their efforts often serve to highlight their lack of infrastructural support. In other instances the underlying mechanisms to get the information they needed were just fragile.

8. Conclusion

In this paper we have explored the needs of one stakeholder in the ecosystem of AMT, the Turkers. We have found several issues that Turkers are experiencing, as well as several workarounds that they are using to work around these issues. These issues seem to indicate breakdowns in the AMT ecosystem, which have negative effects for stakeholders other than just Turkers.

One example breakdown that Turkers both suffer from and contribute to is the velocity of the market. Turkers are playing an active role in accelerating this velocity through their efforts to gain a competitive advantage. This makes it more difficult for new Turkers to onboard, as they may never encounter the most popular HITs. The difficulty onboarding new workers for this labor market has a negative impact both on requesters and Amazon, as new Turkers quit more frequently.

Another example of a breakdown that impacts Turkers is the poor search interface provided by AMT. This poor interface increases the opportunity cost of finding new, higher quality work. There are several effects of this, one interesting one is the requester-brokers who serve as a liaison between the crowd and employers. Crowdfunder is an example broker, who after starting on AMT, has since migrated away from AMT because they were having trouble getting quality results from the platform,¹² which of course presents a problem for Amazon as a stakeholder.

We hypothesize that, at least one possible contributing cause of these breakdowns is a lack of consideration for Turkers, an important stakeholder in the ecosystem. This is problematic in that it is difficult to create a complex, socio-technical system with an incomplete picture of stakeholder needs and concerns. We conclude that there is a lack of consideration for Turkers based on the imbalanced distribution of functionality and the policies of AMT. It seems that Amazon views AMT as a service that they are providing for the requesters who are

¹² <http://www.crowdfunder.com/blog/2014/01/crowdfunder-drops-mechanical-turk-to-ensure-the-best-results-for-its-customers>

paying for the service, which devalues Amazon's relationship with Turkers. In this configuration, the relationships with Turkers and the labor that they provide is somewhat invisible (Star and Strauss 1998), or at the very least taken for granted by Amazon. Perhaps a slightly different framing of AMT, as levying a fee on an exchange of services, may be more realistic. This framing can elevate the importance of the Turker stakeholder group and bring more balance to future functional enhancements. This more holistic view of the ecosystem is needed to properly diagnose and patch the breakdowns in AMT that seem to be developing.

Certainly, if Turkers were ever officially classified as employees of Amazon (not out of question given recent developments with Uber¹³), they would be more likely to be concerned with providing their workers with better tools to complete their work with more speed and accuracy. If we consider that Turkers are at least providing a valuable service to Amazon, it follows that it would benefit Turkers, requesters, and Amazon if their tool set were better tailored for them. Amazon would potentially benefit on several fronts: better connecting quality Turkers with quality requesters; producing higher quality (or at least quicker) work; growing the labor pool through an onboarding path; and increasing their earnings from increased exchanges. Turkers would potentially benefit both financially and by having a better work environment. Requesters would potentially benefit from an increased pool of more efficient workers. Amazon could even use the tools that Turkers have constructed as a source for these future functional enhancements in their platform, in effect the Turkers have already gone about the business of formalizing their requirements for them.

References

- Bederson, Benjamin B.; and Alexander J. Quinn (2011). Web workers unite! Addressing challenges of online laborers. In B. Begole and W. Kellog (Eds): *CHI EA '11. Proceedings of SIGCHI Conference on Human Factors in Computing Systems Extended Abstracts. Vancouver, Canada, 7–12 May 2011*. New York: ACM press, pp. 97–106.
- Boehner, Kristen; Janet Vertesi; Phoebe Sengers; and Paul Dourish (2007). How HCI interprets the probes. In D. Gilmore (Ed): *CHI '07. Proceedings of SIGCHI Conference on Human Factors in Computing Systems. San Jose, California, USA, 28 April – 3 May 2007*. New York: ACM press, pp. 1077–1086.
- Brown, Barry; Stuart Reeves; and Scott Sherwood (2011). Into the wild: Challenges and opportunities for field trial methods. In B. Begole and W. Kellog (Eds): *CHI '11. Proceedings of SIGCHI Conference on Human Factors in Computing Systems. Vancouver, Canada, 7–12 May 2011*. New York: ACM press, pp. 1657–1666.
- Callison-Burch, Chris (2014). Crowd-Workers: Aggregating Information Across Turkers to Help Them Find Higher Paying Work. In: *HCOMP '14. AAAI Conference on Human Computation and Crowdsourcing: Works in Progress and Demonstration Abstracts. Pittsburg, USA, 3–4 November 2014*. AAAI Press, pp. 8–9.

¹³ <http://uberlawsuit.com/OrderDenying.pdf>

Investigating the Amazon Mechanical Turk...

- Chilton, Lydia B.; John J. Horton; Robert C. Miller; and Shiri Azenkot (2010). Task search in a human computation market. In: *HCOMP '10 Proceedings of the ACM SIGKDD Workshop on Human Computation*. Washington DC, USA, 25 July 2010. New York: ACM press, pp. 1–9.
- Felstiner, Alek (2011). Working the crowd: Employment and labor law in the crowdsourcing industry. *Berkeley Journal of Employment and Labor Law*, vol. 32 no. 1, pp. 143–203.
- Gaver, William W.; Andrew Boucher; Sarah Pennington; and Brendan Walker (2004). Cultural probes and the value of uncertainty. *Interactions - Funology*, vol. 11, no. 5, pp.53–56.
- Gupta, Neha; David Martin; Benjamin V. Hanrahan; and Jacki O’Neill (2014). Turk-life in india. In D. McDonald and P. Bjorn (Eds): *GROUP 2014. Proceedings of International Conference on Supporting Group Work*. Sanibel Island, Florida 9–12 November 2014. New York: ACM press, pp.1–11.
- Hanrahan, Benjamin V.; Jutta Willamowski; Saiganesh Swaminathan; and David Martin (2015). Turkbench: Rendering the Market for Turkers. In B. Begole and J. Kim (Eds): *CHI '15. Proceedings of SIGCHI Conference on Human Factors in Computing Systems*. Seoul, South Korea, April 18–23, 2015. New York: ACM press, pp. 1613–1616.
- Hutchinson, Hilary; Wendy Mackay; Bo Westerlund; Benjamin B. Bederson; Allison Druin; Catherine Plaisant; Michel Beaudouin-Lafon; Stéphane Conversy; Helen Evans; Heiko Hansen; Nicolas Roussel; and Björn Eiderbäck (2003). Technology probes: Inspiring design for and with families. In P Baudisch (Ed): *CHI '13. Proceedings of SIGCHI Conference on Human Factors in Computing Systems*. Paris, France, 27 April–2 May, 2013. New York: ACM press, pp. 17–24.
- Ipeirotis, Panos G. (2010). Analyzing the Amazon Mechanical Turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, vol. 17, no. 2, New York: ACM press, pp. 16–21.
- Irani, Lily C.; and M. Six Silberman (2013). Turkopecticon: Interrupting worker invisibility in amazon mechanical turk. In P Baudisch (Ed): *CHI '13. Proceedings of SIGCHI Conference on Human Factors in Computing Systems*. Paris, France, 27 April–2 May, 2013. New York: ACM press, pp. 611–620.
- Khanna, Shashank; Aishwarya Ratan; James Davis; and William Thies (2010). Evaluating and Improving the Usability of Mechanical Turk for Low-Income Workers in India. In A. Dearden, J. Parikh and L. Subramanian (Eds): *DEV 2010. Proceedings of ACM Symposium on Computing for Development December 17–18, 2010, London, United Kingdom*, New York: ACM press, article no. 12.
- Kittur, Aniket; Jeffrey V. Nickerson; Michael Bernstein; Elizabeth Gerber; Aaron Shaw; John Zimmerman; Matt Lease; and John Horton (2013). The future of crowd work. In: *CSCW '13. Proceedings of Computer Supported Cooperative Work and Social Computing*. San Antonio, Tx, USA. February 23–27, 2013. New York: ACM press, pp. 1301–1318.
- Martin, David; Benjamin V. Hanrahan; Jacki O’Neill; and Neha Gupta (2014). Being A Turker. In M. Ringel Morris and M. Reddy (Eds): *CSCW '14. Proceedings of Computer Supported Cooperative Work and Social Computing*. Baltimore US, 15-19th February 2014. New York: ACM press, pp. 224–235.
- O’Neill, Jacki; and David Martin (2013). Relationship-Based Business Process Crowdsourcing. In P. Kotzé, G. Marsdan, G. Lingaard, J. Wesson and M. Winckler (Eds): *INTERACT '13. Proceedings of Human-Computer Interaction, 14th IFIP TC 13 International Conference Cape Town, South Africa, September 2-6, 2013*, Proceedings, Part IV, pp. 429–446.
- Quinn, Alexander J.; and Benjamin B. Bederson (2011). Human computation: a survey and taxonomy of a growing field. In B. Begole and W. Kellog (Eds): *CHI '11. Proceedings of SIGCHI Conference on Human Factors in Computing Systems*. Vancouver, Canada, 7–12 May 2011. New York: ACM press, pp. 1403–1412.
- Salehi, Niloufar; Lilly Irani; Michael Bernstein; Ali Alkhatib; Eva Ogbe; and Kristy Milland (2015). We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In B. Begole and J. Kim (Eds): *CHI '15. Proceedings of SIGCHI Conference on Human Factors in*

- Computing Systems. Seoul, South Korea, April 18–23, 2015*. New York: ACM press, pp. 1621–1630.
- Silberman, M. Six; Joel Ross; Lilly Irani; and Bill Tomlinson (2010). Sellers' problems in human computation markets. In: *HCOMP '10 Proceedings of the ACM SIGKDD Workshop on Human Computation. Washington DC, USA, 25 July 2010*. New York: ACM press, pp. 18–21.
- Star, Susan Leigh; and Anselm L. Strauss (1998). Layers of silence, arenas of voice: The ecology of visible and invisible work. *Computer Supported Cooperative Work (CSCW): An International Journal*, vol. 8, no. 1, pp. 9–30.