Turkish Judge: A Peer Evaluation Framework for Crowd Work Appeals

Edward Cohen, Mukund Venkateswaran, Nivedita Sankar, Chris Callison-Burch

Abstract
We present a crowd-driven adjudication system for rejected work on Amazon Mechanical Turk. The Mechanical Turk crowdsourcing platform allows Requesters to approve or reject assignments submitted by Workers. If the work is rejected, then Workers aren’t paid, and their reputation suffers. Currently, there is no built-in mechanism for Workers to appeal rejections, other than contacting Requesters directly. The time it takes Requesters to review potentially incorrectly rejected tasks means that their costs are substantially higher than the payment amount that is in dispute. As a solution to this issue, we present an automated appeals system called Turkish Judge which employs crowd workers as judges to adjudicate whether work was fairly rejected when their peers initiate an appeal. We describe our system, analyze the added cost to Requesters, and discuss the advantages of such a system to the Mechanical Turk marketplace and other similar microtasking platforms.

Introduction
In this paper, we propose an add-on to the Amazon Mechanical Turk crowdsourcing platform to give Workers a way of appealing rejected work, and to simplify the Requester’s job of adjudicating their appeals by allowing the Requester to hire other crowd workers to act as peer reviewers of the rejected work. This allows the Requester to overturn the rejection or uphold it with minimal effort. It reduces the burden placed on Requesters to manually review appeals, and provides for the fairer treatment of Workers.

Amazon Mechanical Turk (AMT) is a crowdsourcing platform developed by Amazon which allows businesses and individuals (Requesters) to have microwork completed by Workers (sometimes called Turkers) on the platform. Tasks uploaded by a Requester to the Amazon Mechanical Turk marketplace to be completed by Turkers are called HITs (Human Intelligence Tasks). AMT allows Requesters to review HITs submitted by Workers. Requesters may approve the HITs, in which case the Worker is paid, or they may reject the HITs, in which case the Worker is not paid.

Beyond wasted time and not receiving compensation for their work, a Worker is additionally penalized since a rejection diminishes their rating in AMT’s reputation system. Each Turker has an associated score based on their past approval rate. Requesters restrict their work to Workers whose approval rate is above a specified threshold. Whenever a Worker receives a rejection their approval rate decreases, which limits the work they are able to access on the platform.

Requesters are able to reject HITs submitted by Turkers for any reason whatsoever, and AMT provides no mechanism for Workers to appeal the rejection beyond emailing the Requester. This brings obvious concerns of fairness. Workers are aware of this issue and in a 2019 survey, 69% of Workers cited unfair rejections in their top three grievances with the platform (Whiting, Hugh, and Bernstein 2019). Turkers have means to label and make others aware of unfair Requesters through either TurkOpticon (Irani and Silberman 2013) or popular forums like Turker Nation. On TurkOpticon, Turkers can rate Requesters on their “fairness”. This attribute is a 1 to 5 scale for how fair a Requester is in approving or rejecting work. Low ratings signal to Workers that their peers have had issues with a Requester in the past and they should exercise caution when considering accepting work from that Requester.

It must be noted that some of the rejections on the platform are just, and weeding out malicious Workers is a necessary tool for Requesters. However, Workers often see their rejections as unfair (McInnis et al. 2016). Mechanical Turk has built-in functionality allowing Requesters to overturn rejections, but Workers have little recourse for asking for this. Mechanical Turk supports Workers emailing Requesters directly asking them to overturn the rejection. However, Requesters rarely read or respond to these because of the time costs associated with a manual resolution of the grievance (Irani and Silberman 2013). Therefore, any proposed solution to this issue must be minimum effort to Requesters.

In the search for this solution, we devised an automatic appeals mechanism for AMT in which a Worker whose HIT was rejected has the option to submit an appeal of that rejection onto the marketplace. The mechanism posts the HIT in question and an explanation provided by the appellant...
on the AMT marketplace for other Turkers to adjudicate on whether the rejection was fair or unfair. The buy-in from requesters is that they upload a file containing their HIT results and check in to see the status of their rejected HITs.

We introduce Turkish Judge, an implementation of this mechanism for Turkers to appeal their rejected HITs. Turkish Judge seeks to make the work denial process fairer by incorporating an experienced third party in the form of other Turkers who have completed similar tasks.

In this paper, we describe the Turkish Judge mechanism and implementation in detail and provide an empirical analysis on the adjudication quality of such a mechanism. We conduct an economic analysis of the cost saved by Requesters in time-value in order to examine Requester incentives in the use of such a mechanism. Finally, we provide suggestions as to further analysis and experimentation to be conducted in the analysis of a crowd-based appeals system and detail planned add-ons to our implementation prior to a full deployment of the system. Our code for the system is currently available at [ANON].

Description

Need

There are many different use cases for Requesters on AMT. A common one is for researchers to deploy hundreds or thousands of tasks to individual Workers, either for social science surveys or to gather large amounts of labels to train machine learning system. For such large-scale task deployments it is common for Requesters to design automatic quality-control mechanisms to determine whether to approve or reject each submission, since manual review of each submission would be too time consuming. Quality control mechanisms often use attention checks or gold standard questions embedded in the task. Workers whose responses differ from the expected answers for these questions have their submissions rejected. However, there are important ramifications that come with rejecting work. A Turker whose work has been rejected not only receives no payment for the submission, but also suffers a negative weighting to their reputation on the platform, which impedes them from participating in other tasks, limiting future earnings on the platform. A low enough reputation can even get a Turker banned from the platform. There is little recourse for Workers to escalate this issue beyond asking for reconsideration. As Bederson and Quynn discuss, unfairly rejected work is the top priority problem for Mechanical Turk Workers, given the financial and reputation implications in both the present and the future. Based on a analysis of posts on Turkerculture, they find that crowd workers “do not feel that they are adequately supported by the existing infrastructure” (Bederson and Quynn 2011). Nothing to do with petitioning rejected work on Mechanical Turk has changed since then.

When a Turker submits a grievance request regarding their rejected work, the Requester must read the petition in full, manually search through the HIT batch data for the corresponding row entry, assess its quality, and return a final judgement. This process that takes a nontrivial amount of time. Given that the median hourly wage for Turkers is around two dollars an hour (Hara et al. 2017), the Requester incurs more cost reviewing the petition than they would have paid originally. Irani captured this sentiment in a quote from a Requester who said “[a Requester] cannot spend time exchanging email. The time you spent looking at the email costs more than what you paid [the Worker].” (Irani 2015). This imbalance means that the petitioning Turker frequently receives no response, much less payment and a reputation increase.

As a result, a mechanism that removes the time burden associated with Requesters manually reviewing appeals. A mechanism which allows Requesters to adjudicate appeals quickly and at low cost could enable a marketplace that is both easy for Requesters and fairer to Workers.

Current Status

In order to have a rejected assignment reversed, a Turker must send a message to the corresponding Requester. The Requester is not obligated to reply. If a Requester chooses to, they may then review the HIT in question and manually reverse the rejection for the Turker. There are three heuristic methods for adjudicating petitions which Requesters typically employ: ignoring the grievance request submitted by the Turker, manually taking the individual HIT into consideration, or reversing the rejections without review.

Automatically Reject As stated above, Requesters often ignore the grievance emails they receive, effectively automatically rejecting all petitions. Automatically rejecting work requires no extra time taken to consider the case behind each assignment of submitted work and requires no extra payment from the Requester. However, in the case that the work was erroneously rejected, this option runs the risk of Workers negatively posting about their experience on TurkOpticon. This may impact the pool of Workers who opt into that Requester’s tasks in the future, and thus their ability to receive high quantities of quality work.

Additionally, there are ethical concerns to be taken into account. Rejected work has a long-standing negative impact on the wages, as well as the reputation of Workers and their forward ability to work on AMT. In fact, the cost implications may be of lesser concern than the reputation considerations. Many Workers submit grievance requests asking for a Requester to approve previously rejected work but forego payment because the reputation concerns are of greater importance (Salehi et al. 2015).

Individual Consideration Requesters may also choose to individually adjudicate grievance requests to maximize upon fairness concerns. This is less time efficient than automatic rejection of petitions, and a Requester may have to send initially withheld payment. This, however, ultimately leads to a fairer outcome. From an ethical perspective, a Requester who carefully considers each submitted petition will in effect pay Turkers who completed the task in good intentions and warn other Requesters about Turkers who submitted poor quality work. There is some measure of unfairness if the low-quality work can be attributed to confusing HIT design, and the Requester fails to take this into account.
Reversing Rejections Without Review Some Requesters choose to reverse rejections whenever a Worker complains without actually inspecting the submitted work. This makes sense economically, since the amount of time it takes a Requester to review the petition is worth more than the amount in dispute. And, the Requester does not want to risk their fairness rating with potentially erroneous rejections. This strategy is potentially detrimental to the AMT reputation system because it relies on Requesters accurately judging the quality of Workers’ submissions. Requesters who approve all work, or who reverse rejections without review artificially inflate the reputations of workers who do their tasks.

Reasons for Rejection
There are a number of reasons, fair and unfair, that a Requester rejects a Turker’s work. We provide a list that describes a few of these reasons. Note that fair rejections are from the perspective of the Requester and unfair rejections are from the perspective of the Turker.

Fair rejections:
1. The Turker used a script to complete the HIT.
2. The Turker manually completed the work but did so randomly, effortlessly, or untruthfully.
3. The Turker performed poorly on gold standard quality checks indicating low aptitude for the HIT.
4. The Turker did not follow instructions.

Unfair rejections:
1. The instructions were vague or unclear.
2. Faulty HIT design which includes:
   (a) unanswerable quality checks
   (b) not enough time to complete the HIT
3. Malicious rejections for no reason.

Of the reasons listed, the more frequent reason for a rejection is because of quality or attention checks. Requesters frequently use a gold standard for measuring the quality of submitted work but do not always ascertain the quality of the checks themselves. An example from a subreddit dedicated to Mechanical Turk, r/mturk, shows a HIT that asked Turkers “How many fatal heart attacks have you suffered in the past week?” (bold was in original) and did not provide 0 as an answer choice. Whether this was a miswritten question or a poorly designed attention check is unclear, but Turkers are afraid of getting rejected in instances like this.

Related Work
There have been numerous investigations into the grievances of Turkers through surveys on the platform and analysis of forums for Turkers. Most of the previous work has highlighted the issue of Workers’ wages on the platform. Specifically, the far-below minimum wage that Turkers receive hourly (Hara et al. 2017). The connection between qualms about wages and unfair rejections has been noted and emphasized (Bederson and Quinn 2011). Additionally, unfair rejections as a standalone issue has been reported as a major concern. In reports dating back to the creation of TurkOpticon, 35 out of 67 Turkers reported they felt their work was frequently unfairly or arbitrarily rejected (Irani and Silberman 2013). More recently, unfair rejections have been reported in the top three issues 69% of Turkers have with the site (Whiting, Hugh, and Bernstein 2019). This distrust causes some Workers to purposefully submit low-quality work, and some Requesters are put on blacklists of Workers (Bederson and Quinn 2011).

The literature on potential solutions is written in the context of design considerations for the platform beyond just unfair rejections. (Bederson and Quinn 2011; Martin et al. 2014). Generally, Turkers are identified as an invisible workforce with calls being made for more transparency and communication on the platform (Martin et al. 2014). Toward this end, services like TurkOpticon and Crowd Workers, as well as platforms like TurkNation provide Workers with information into Requesters and the marketplace (Irani and Silberman 2013; Callison-Burch 2014). On the issue of rejections specifically, Bederson and Quinn suggest that Requesters should have to at least email Turkers with an explanation for why their work was rejected (Bederson and Quinn 2011). Turkers have suggested three options: they are given the chance to redo the work, the Requesters should not be allowed to keep rejected work, and similar to Bederson and Quinn, that Requesters are obligated to elucidate a reason for the rejection (Felstiner 2011).

The most thorough investigation and tangible solution was Daemo (Gaikwad et al. 2015). Daemo was an alternative crowd work marketplace designed by the crowd that encouraged task requesters and workers to cooperate on refining tasks. This cooperation was mutually beneficial. Workers did not have to worry about unclear instructions or faulty questions. Requesters developed trust for the people completing their task so they did not have to worry about malicious workers. At this point, the Daemo site is no longer live, but we seek to continue their ideals of a more balanced requester-worker relationship.

The worker review platform TurkerView released a rejection dispute tool, TurkerView Bridge this past February. The feature is built into the Mechanical Turk worker interface. Turkers can submit a text, image, or video reason for why the work should not have been rejected. This appeal gets sent to the requester for them to review. TurkerView Bridge provides information for the requester on overturning rejections and a properly formatted csv for one to upload to reverse rejections. This solution reduces the friction in the rejection dispute process but still relies on requesters to take the time to review disputed rejections.

Turkish Judge: the Mechanism
The Turkish Judge mechanism is an automatic appeals system for the Mechanical Turk platform. It allows for Turkers who feel their work was unfairly rejected to submit an appeal of the rejection. Turkers must provide an explanation or supporting message along with their appeal. This message,
in combination with the original completed work, comprises an appeal. The appeal information is posted onto the AMT marketplace for other Turkers to adjudicate. In the case that the adjudicating Turkers decide that the rejection should be overturned, the payment for the original task is given to the rejected Worker and the penalty to their approval rating is undone.

Requesters
After a batch is completed and reviewed by a Requester, the Requester should upload the batch information to Turkish Judge for adjudication. Here, we have access to the relevant information necessary for the adjudication task including HIT IDs, Worker IDs, the HIT in question, and whether the task was accepted or rejected. This information is stored in our database to allow rejected Workers to appeal their HITs. Once a Turker submits an appeal and the task is adjudicated, the Turker and the Requester will both be notified so that the Requester can act correspondingly to overturn the rejection the Turker if the rejection is deemed unfair.

This allows for Requesters to spend much less time dealing with messages from Turkers who were rejected and introduces an expert third party to analyze the completion of the HIT.

Additionally, we plan on implementing an interface through which Requesters can directly upload completed HITs for adjudication, without the need for the corresponding Turker to appeal. This will allow for Requesters to use the platform as a quality control mechanism rather than other automated methods, or to use the platform to confirm that suspected malicious HITs should be rejected prior to actually rejecting them.

Therefore the only work required from the Requester is to upload completed HITs, and overturn rejections if and when they are deemed unfair. Both of these take brief amounts of time and thus are minimum effort options to Requesters.

Appealing a HIT
In order to appeal a rejected HIT, Workers provide the ID of the HIT in question along with their Mechanical Turk Worker ID in order to identify the corresponding HIT. The Turker will be prompted to enter an explanation for their completion of the HIT describing why the rejection should be overturned. Once a Worker submits an appeal, an adjudication task is created for 3-5 other Turkers to complete.

Adjudicating a HIT
An adjudication task is comprised of the original HIT in question, an optional appeal message from the Turker, the rejection message from the Requester, as well as some instructions on how to adjudicate the HIT as objectively as possible. The Judge who is given this task will be able to respond either that the rejection should be upheld or that it should be overturned. We aggregate the responses that we get from the Judges to make a decision on the task in question.

Once adjudications are collected, we simply use a majority vote over the 3-5 adjudications provided in order to aggregate and make a verdict.

Overturning Incorrectly Rejected HITs
On Mechanical Turk, Requesters have the option of uploading a CSV file with “Approve” and “Reject” columns to overturn rejected HITs. After a verdict is reached on all appealed HITs in a batch, the Requester will be able to download a csv file containing the verdict for each HIT. A Requester will simply be able to upload this csv file to Mechanical Turk in order to reverse the rejection on the overturned HITs. This greatly accelerates the Requester workflow from manual adjudication, as they simply have to upload one csv file rather than individually dealing with each Worker grievance.

An Empirical Analysis of Adjudication Quality
A prominent concern with such a mechanism is adjudication quality. More specifically, how can effortful and truthful adjudication be incentivized? One concern is that because Turkers are predisposed to feel Requesters rejections are unfair, perhaps Turkers would simply side with their peers when adjudicating HITs. This issue is of the utmost importance in the usefulness of our system, since there is no use in a corrupt judicial system.

We simulate a batch of rejected work to analyze how Judges adjudicate appeals, in order to provide an empirical estimate of the quality of our adjudication mechanism. Each of the appeals has a ground truth label for whether it should be overturned or not. We believe that Turkers will effectively determine whether or not a rejected HIT was actually properly completed because they are experts in completing HITs. Using the gathered data, we seek to measure the productivity and behavior of Workers on the adjudication task.

Methodology
In order to simulate rejected HITs, we use a text classification task in which Workers are asked to identify adjectives which describe a given attribute. Gold standard labels are produced in accordance with the accuracy of the generated answers: we generate HITs in which over 75% of the questions were correctly answered as HITs that were properly completed and less than that as HITs that should be rejected. We generated 48 HITs that should be upheld in rejection and 50 HITs whose rejection should be overturned. Along with these generated HITs, we provide an appeal explanation as implemented in our actual system; an example being: “I tried to do the best I could, but the set up was a little more than confusing.” Judges were instructed to read the original HIT in full, consider the explanation, and submit whether they think the rejection was fair or unfair. Five unique Judges completed each HIT. An example depiction of the original rejected HIT completed is in [Figure 1].

Quality Control
Maintaining a high standard of quality control for Workers who act as the Judges was of utmost importance for our objective. Because this appeal task required reviewing the quality of other non-trivial tasks, it was important that Judges understood the original task and were able to
skillfully evaluate the correctness of said task. In order to maintain these quality control standards, we employed the following restrictions to separate eligible Mechanical Turk Workers from non-eligible Workers:

1. Number of HITs Approved is greater than 100
2. HIT Approval Rate for all Turker’s completed HITs is greater than 90 percent
3. Location is in a country with English as the primary language

Criteria (1) and (2) made it more likely for Workers to return higher-quality results based on their performance on other previous HITs. Criterion (3) was important due to the fact that we wanted Workers to have a good grasp on English and various adjectives; having Workers hail from countries with English as the primary language increased the likelihood of this standard.

Amazon’s Mechanical Turk platform made it easy to restrict the completion of our tasks solely to Workers who met the aforementioned criteria. The platform also includes numerous other options for maintaining quality control, but these options come at a premium. Because of our desire to analyze the economic feasibility of our system, we decided to only use the no-cost filters.

Implicitly, our system functions without explicit quality control methods beyond Worker qualifications by placing trust in the Workers on Mechanical Turk, specifically the ones who have proven to be skilled at tasks, as those are the ones who complete judgements. By showing the complete instructions and examples for the original HITs our highly qualified Workers get an in-depth understanding of what was asked on the original task, and thus should have an idea of how to complete it.

Results

We summarize the results of our adjudications tasks in the tables above [Figures 2, 3].

Note that each verdict is representative of five Workers, making each individual decision made by the judges more statistically significant than if adjudicated simply by one or three Workers.

Analysis

To quantify the quality of adjudication through the formulated crowd based appeals system, we focus on the precision and recall of overturning rejected HITs.

In determination of true labels for HIT completion, a generous threshold for properly completed HITs might be an accuracy of .75. Using this as the true labels for our empirical experimentation, we find that there are very few false positive predictions made by the Judges. This shows that Judges very rarely reverse rejections on improperly completed HITs.

On the other hand, we find quite a few false negative predictions. This shows that our Judges frequently upheld rejections when we would expect the rejection to be overturned according to our performance threshold. We break this down further in [Figure 3] to show the threshold of accuracy expected by Judges when overturning a rejection. With an accuracy even as high as .83, we find that Turkers still uphold rejection on the majority of the appeals. Only on HITs completed with perfect accuracy do we find that Judges overturn a majority of the HITs.

These results indicate Judges’ proficiency in the analysis of HIT completion quality as we can see that the rejection reversal rate increases monotonically with the accuracy of HIT completion. Moreover, these results indicate that Judges
Economic Analysis and Incentives

As previously discussed, the environment surrounding Worker grievance requests is important to address from the perspective of both the Requesters and the Turkers. Under the current system, Requesters face significant time-cost burdens, which incentivizes less careful behavior surrounding each request. Turkers, on the other hand, have a deep stake in having these issues addressed properly in a quick and fair manner for both short-term wage concerns as well as longer-term concerns such as reputation on the platform.

Requesters typically have three approaches to dealing with Worker grievances: ignoring Worker grievance requests, individual consideration of each grievance, and blanket reversal of rejections. The first method contributes to the power imbalance on the platform, whereas the second carries the aforementioned burden of time-costs, and the third undermines AMT's reputation system. Through the Turkish Judge platform, Requesters have a viable alternative to dealing with Worker grievances. Turkish Judge alleviates the time burden that Requesters face, but it comes at some cost since the Judges must be paid to adjudicate appeals request.

In this section, we estimate the cost of running Turkish Judge based on the payments to Judges, and varying the rejection/appeal rates on the Requester's original task.

Requester Adjudications

In understanding the economic viability of the Requester-based options, the time taken by the Requester to manually adjudicate and overturn a HIT was estimated at ten minutes. Each minute of the requester's time was valued at $.50 representing a typical salary paid to a research scientist in the United States. On top of this, the Requester must pay the original wage in question to the Turker along with the 20% royalty taken by the Mechanical Turk platform.

Automatically Reject  As previously stated, this option is of no cost to the Requester as this option implies simply ignoring grievance requests submitted by Turkers. Though this is of low monetary and time costs to the Requester, there are many ethical issues with such an option especially considering task acceptance and rejection is often automated in the first place.

Individual Consideration  From a cost perspective, manually addressing and considering the merits of each single request is estimated to take the Requester ten minutes on average. We consider rejection rates ranging from 1.0% to 20.0% and appeal rates on these rejections to range from 50.0% to 100.0% against a total number of 1000 HIT submissions. A full table of the time costs based on the Requester time-value estimate can be found in [Figure 4].

Analysis of these results shows that the time costs of manual considerations of each grievance requests may fall as low as $25.00 in the case of a minimal rejection and appeal rate. On the other hand, the time costs may be as high as $1000.00 in the case of the maximal rejection and appeal rates considered. Note that these costs do not take the actual wages paid to the original Turker or royalties paid to Mechanical Turk into account.

Reversing All Rejections Without Reviewing  The cost to the Requester of reversing all rejections without reviewing them is simply the original payment to the Worker. Though in many cases this cost is less than that of the time cost associated with manual adjudication, this practice has other ramifications. As Workers often qualify for tasks based off of their approval rating, artificial inflation of the approval rate through this practice can lead to lower quality work for all Requesters on the platform. This undermines the Worker reputation system on AMT.

Cost of Crowd-based Adjudication with Turkish Judge

The previous section addresses the cost to Requesters for handling Turkver grievance requests by themselves. However, an alternate approach outlined by this paper is to outsource the adjudications of tasks to crowd workers to form a sort of judicial system. This method allows us to address fairness concerns along with Requester concerns of time burden for manual adjudication. Though the crowd workers serving as judges for this task require payment, essentially all other components of the process are automated.

To understand the economic viability of this option, we assume the adjudication costs to be 60 cents per task. This cost is comprised of a ten cent payment to each adjudicating Worker and a 20% royalty paid to the Mechanical Turk platform. We use the same aforementioned framework of 1000 HITs and the same ranges of rejection and appeal rates. A full table of the estimated costs to Requesters for the use of such a service can be found in [Figure 5].

Analysis of these results shows that the cost of the service to the Requester in order to pay the adjudicating Workers falls as low as $3.00 in the case of a minimal rejection and appeal rate of 1.00% and 50.00% respectively. In the maximal case of a 20.00% rejection rate and a 100.00% appeal rate, the cost of the service is $120.00. Again, note that these costs do not take into account the wages paid to the original Turker or the royalties paid to Mechanical Turk for the original HITs.

This shows that using our model for a batch of 1000 HITs, the Requester is able to observe anywhere from $22.00 to $880.00 in time cost savings under the assumptions made in this analysis.

We also briefly consider the case in which the Requester uses this platform as a second opinion for HITs they intend to reject before the rejections are sent to Workers. We note that this case essentially represents a 100% appeal rate by Turkers as we are simply uploading all tasks that would be rejected by the Requester to the mechanism. Though this initially seems costly, it is important to note that the Requester does not have to spend as much time developing accurate quality control tools for different tasks. Tools which are simply able to detect malicious Turkers with a high rate of recall (such as embedded gold standard answers) are sufficient as Turkers have empirically shown to be precise in determination of HIT completion quality.
Figure 4: Table of cost estimates for Requesters to manually adjudicate on a batch of 1000 HITs, valuing their time at $30/hour and estimating 5 minutes per adjudicated task. The values represent the dollar amount in the Requester’s time cost for the corresponding rejection rate, appeal rate pair.

<table>
<thead>
<tr>
<th>Rejection rate / Appeal rate</th>
<th>50% appeal rate</th>
<th>60% appeal rate</th>
<th>70% appeal rate</th>
<th>80% appeal rate</th>
<th>90% appeal rate</th>
<th>100% appeal rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1% rejection rate</td>
<td>$25.00</td>
<td>$30.00</td>
<td>$35.00</td>
<td>$40.00</td>
<td>$45.00</td>
<td>$50.00</td>
</tr>
<tr>
<td>5% rejection rate</td>
<td>$125.00</td>
<td>$150.00</td>
<td>$175.00</td>
<td>$200.00</td>
<td>$225.00</td>
<td>$250.00</td>
</tr>
<tr>
<td>10% rejection rate</td>
<td>$250.00</td>
<td>$300.00</td>
<td>$350.00</td>
<td>$400.00</td>
<td>$450.00</td>
<td>$500.00</td>
</tr>
<tr>
<td>15% rejection rate</td>
<td>$375.00</td>
<td>$450.00</td>
<td>$525.00</td>
<td>$600.00</td>
<td>$675.00</td>
<td>$750.00</td>
</tr>
<tr>
<td>20% rejection rate</td>
<td>$500.00</td>
<td>$600.00</td>
<td>$700.00</td>
<td>$800.00</td>
<td>$900.00</td>
<td>$1,000.00</td>
</tr>
</tbody>
</table>

Figure 5: Table of cost estimates for using our Turkish Judge tool to adjudicate a batch of 1000 HITs. All other values represent the dollar amount in costs to the Requester to pay Judges in our crowd-based appeals system.

<table>
<thead>
<tr>
<th>Rejection rate / Appeal rate</th>
<th>50% appeal rate</th>
<th>60% appeal rate</th>
<th>70% appeal rate</th>
<th>80% appeal rate</th>
<th>90% appeal rate</th>
<th>100% appeal rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1% rejection rate</td>
<td>$3.00</td>
<td>$3.60</td>
<td>$4.20</td>
<td>$4.80</td>
<td>$5.40</td>
<td>$6.00</td>
</tr>
<tr>
<td>5% rejection rate</td>
<td>$15.00</td>
<td>$18.00</td>
<td>$21.00</td>
<td>$24.00</td>
<td>$27.00</td>
<td>$30.00</td>
</tr>
<tr>
<td>10% rejection rate</td>
<td>$30.00</td>
<td>$36.00</td>
<td>$42.00</td>
<td>$48.00</td>
<td>$54.00</td>
<td>$60.00</td>
</tr>
<tr>
<td>15% rejection rate</td>
<td>$45.00</td>
<td>$54.00</td>
<td>$63.00</td>
<td>$72.00</td>
<td>$81.00</td>
<td>$90.00</td>
</tr>
<tr>
<td>20% rejection rate</td>
<td>$60.00</td>
<td>$72.00</td>
<td>$84.00</td>
<td>$96.00</td>
<td>$108.00</td>
<td>$120.00</td>
</tr>
</tbody>
</table>

Turkish Judge: the System/Implementation

Turkish Judge is implemented as a simple Flask web application. There is a single homepage that Requesters and Turkers interact with. Requesters can either sign in or follow a link to create an account from the home page. Turkers can enter their Worker ID and HIT ID to submit an appeal or check the status of a published appeal once the corresponding Requester has uploaded a batch. Each of the user interfaces is displayed in [Figure 6].

Requesters

Requester Interface  Once a Requester has signed in, they are brought to the Requester dashboard. This displays the batches they have uploaded, as well as a form to upload another batch either as a csv or tsv. When a Requester uploads a batch, each rejected assignment is stored as an unappealed HIT in our database and the Question and Answer XMLs are aligned to recreate the HIT as the Worker completed it. We provide a link to the completed HIT on our website so that both the Requester and the Worker are able to access it through the adjudication process. The Requester is brought back to their dashboard with the new batch displayed in their uploaded batches. Each batch they have uploaded includes a link to take the Requester to a batch page that shows them statistics and statuses of the assignments of that batch. A Requester can click on an assignment to see in depth information about the appeal, including a link to the completed assignment and the current vote count for submitted adjudications.

Requester Workflow  After rejecting HITs from a batch, the Requester will simply have to add a few sentences to their rejection message along the lines of “We are using the Turkish Judge platform to arbitrate Worker grievances on this batch. Once the batch is submitted in the next 24-48hr, you will be able submit an appeal for this HIT at [url].” Then, the Requester will simply have to upload the batch csv to the Turkish Judge platform to allow for Turkers to submit appeals on their rejected HITs.

Once a Requester has created an account, they can simply upload a csv file containing information about the batch. After parsing and uploading the rejected HITs of the batch to our database, Turkers will be able to submit appeals on the corresponding HITs if they wish to do so. After adjudications are complete, the Requester is able to download a csv file from the site with information on which HITs to reverse rejections on. This heavily simplifies the Requester workflow from dealing with Worker grievances individually and manually.

Turkers who Want to Appeal a Rejection

Turker Interface  Once a Worker submits their Worker ID and HIT ID corresponding to a rejected HIT uploaded by a Requester, they are brought to the Worker dashboard. On this page, the Worker has a link to see their completed HIT and if they haven’t submitted an appeal can submit one. This page is where a Worker writes the explanation included in their appeal. Workers can also optionally enter an email to receive notifications. After an appeal is submitted, the corresponding appeal HITs are created and uploaded to Mechanical Turk to be adjudicated. If a Worker returns to this page after an appeal is submitted, they can view the current status of their appeal and change their email if they wish to do so.

Turker Workflow  When a Worker receives a rejection from a Requester who uses Turkish Judge, they should be notified that a Requester is using Turkish Judge to arbitrate Worker grievances in the rejection message. Once the Requester uploads the batch HITs, Turkers can go to the provided URL and submit an appeal. After a verdict is reached, Turkers receive a notification at their email address, if they provided one, and the Requester should act accordingly once the entire batch is adjudicated.
Figure 6: The Requester dashboard and the page to submit an appeal.

Judges

An appeal HIT consists of the original submitted HIT completed by the appealing Turker, an optional message from that Turker, as well as instructions for adjudicating the HIT. A Judge presented with this HIT should analyze the quality of HIT completion, and respond with either “Yes, this task should have been rejected.” or “No, this task should not have been rejected.”

Future Work

Our suggestions for future work include analysis of a variable payment mechanism to further incentivize truthful reporting and further experimentation of the effects of appeal HIT design on the quality of adjudications.

We note that work has been completed in the area of mechanism design to formulate a variable payment mechanism in order to elicit truthful responses from Workers. A formulation of such a mechanism specific to a crowd-based adjudication system may further improve the truthfulness properties of such a platform.

Further experimentation of the effects of appeal HIT design may also improve the quality of adjudications. Notably, experimentation with Worker qualifications such as requiring that to adjudicate a rejection a Worker must have completed a HIT in the original HIT batch. We hope that this experimentation leads to even better adjudication quality.

We also plan on implementing more built-in Requester tools prior to a full deployment of the web application. These include a tool to suggest tip amounts to give Workers, a tool to allow Turkers to redo a HIT to reverse a rejection, and adding other relevant batch statistics on task completion to the Requester dashboard.

Conclusion

Our analyses show that both Turkers and Requesters have incentives to use a crowd-based adjudication mechanism to absolve Worker grievances. Specifically, the relationship between Turkers and Requesters improves on a more equitable platform. This potentially leads to higher quality work and greater trust between Workers and Requesters. Turkers can now appeal rejected work to be adjudicated by a third-party with experience completing HITs, solving a prominent Turker concern, while Requesters are able to accurately and fairly adjudicate petition rejections with minimal effort.

In our empirical analysis, we showed that Turkers are competent and fair in the determination of HIT quality and that they in fact seem to enforce higher standards of HIT completion quality than we had expected. This holds promise for the accuracy and fairness of the Turkish Judge platform.

Through an economic analysis, we examined the costs and consequences of such a mechanism to Requesters compared to their previous alternatives, namely manual consideration and auto-rejection of Worker grievances. We calculate that compared to manual evaluation for a batch of 1,000 HITs, a Requester can save anywhere from $22.00 to $880.00 in time cost under the time-value, rejection rate, and appeals rate assumptions made in our economic analysis.

Having shown the quality of the platform through empirical analysis and economic analysis, we are confident that Turkish Judge is a sound solution and will achieve our larger goal of creating a more equitable marketplace.

References


Felstiner, A. 2011. Working the crowd: Employment and


