Turkish Judge

Cost and Benefit Analysis of a Self Sustainable Appeals System for Mechanical Turk

Peter Chou (choupete)
choupete@seas.upenn.edu

Edward Cohen (edcohen)
edcohen@seas.upenn.edu

Talia Statsky-Frank (tstat)
tstat@seas.upenn.edu

Mukund Venkateswaran (mukundv)
mukundv@seas.upenn.edu

Abstract

We aim to analyze the costs and benefits associated with a self-sufficient appeals system for Mechanical Turk. In particular, we seek to determine whether appeals help in the determination of HIT quality and whether or not Mechanical Turk workers will act honestly in the determination of rejection of a HIT. This will help us determine whether self-sufficient appeals system for Mechanical Turk is even feasible. Additionally, we provide an economic estimation for the value of our service. Finally, we provide suggestions for implementations and further work in the area.

NETS 213: Crowdsourcing and Human Computation

May 2019

Code: https://github.com/tstat1996/nets213_turk_judge

1 Basic Project Information

1.1 One Sentence Description

Turkish Judge analyzes the fairness, costs, and benefits of an automatic appeals system for rejected HITs.

1.2 Problem Formulation

The Turkish judge seeks to improve the fairness of the Mechanical Turk marketplace by creating a self-sustainable appeals system for rejected HITs. Turkers who believe that their work was unfairly rejected will be able to submit an appeal that will be judged by other Turkers. Being the experts in completing HITs, we believe that Turkers will be able to effectively determine whether or not a rejected HIT was actually properly completed. Using the gathered data, we seek to answer the questions: "Are Turkers effective in the determination of HIT quality?" and "How do written appeals affect Turker's ability to determine HIT quality?" Finally, in order to show feasibility, we plan on providing an economic estimation of costs and value that this service would bring to requesters and the Mechanical Turk marketplace.

1.3 Related Work

We found a thread on mturkcrowd about contacting requesters who reject a HIT to ask them to reconsider. Additionally, in 2016, Nivedita Sankar did an analysis on a similar automatic acception or rejection system for Mechanical Turk for her senior design at Penn.

1.4 Type of Project

Our project constitutes a social science experiment with the crowd, with further potential to be a tool used on Amazon Mechanical Turk.

1.5 Team Efforts

The main focus of our team's effort was conducting an in depth analysis of data gathered from Mechanical Turk.

1.6 Methodology

The first step in our project was to collect completed HITs from Mechanical Turk that had an objective answer for whether or not they were completed correctly. For this, we used Jie's data from her Adjectives and Attributes Matching Task that we did Quality Control analysis on for the class. Once we had the HIT data we published two batches on Mechanical Turk, one batch included an appeal from the worker appealing the rejection, and one did not. We generated these appeals ourselves and phrased them as "I did the best I could, but I couldn't do it." Every HIT in each batch was to be completed by five unique workers from the crowd. Workers were simply asked to review the original HIT instructions included with the work completed by a worker and determine whether the rejection was fair or unfair. With the HITs completed our work that required a crowd was finished and we moved on to our automated analysis.

From the data we collected we created two labels for each HIT we posted, one that is the actual "FAIR" or "UNFAIR" label based on whether the worker in the original HIT got at least five out of six of the quality control correct, and the other was our worker's label done using a majority vote on the five workers who reviewed the appeal. After automatically generating labels, we analyzed our results for Precision, Recall, and F score, and did an economic analysis on the feasibility of scaling up this project to review appeals for batches of 1000 HITs.

1.7 Link to Video

https://drive.google.com/a/seas.upenn.edu/file/d/1GMsxWlVYEVLohVwdTmXZ4Xnn02Sp8gi0/view?usp=drivesdk

2 The Crowd

The members of our crowd consisted of workers on the Amazon Mechanical Turk platform. Because Mechanical Turk is both well-established and popular, recruiting participants was as simple as posting the task on the platform. Similar to most other tasks on Mechanical Turk, our task incentivised participation by paying workers a small amount (two cents) per task. That being said, not all workers were able to complete our tasks; we employed various quality control measures in order to better ensure valid results (see section [6]: Quality Control). Overall, our crowd ended up consisting of 56 unique participants; 34 unique participants worked on tasks with appeals, and 40 unique participants worked on tasks without appeals (implying that unique overlap between different-type tasks occurred).

3 Incentives

The crowd is incentivized to complete our hits by monetary compensation and the opportunity to contribute to undoing the power imbalance that exists between requesters and workers on Mechanical Turk.

Our crowd workers, being workers on the platform, are very likely to have experienced the frustration of being unfairly rejected by a requester, as well as the penalty that comes with the rejection both in pay and lifetime approval rating.

Currently, there exist solutions such as Turk Opticon and reddit.com/r/TurkerNation for workers to warn about bad requesters, however this does not directly get to the root of the issue. Turkers have a direct interest in preventing workers from working with unfair requesters, but currently cannot do anything if the work is already done. Through our HITs, we propose a model through which Turkers will be able to vote on whether or not a HIT was properly done. Our workers will not only want to contribute to reduction of unfair rejections on the platform, but are, generally, the experts on knowing whether a HIT was properly done. Assuming a majority of workers honestly complete their work on the platform, our model will yield a fair appeal systems for unfair rejections to possibly overturned. We did not perform any analysis comparing different incentives.

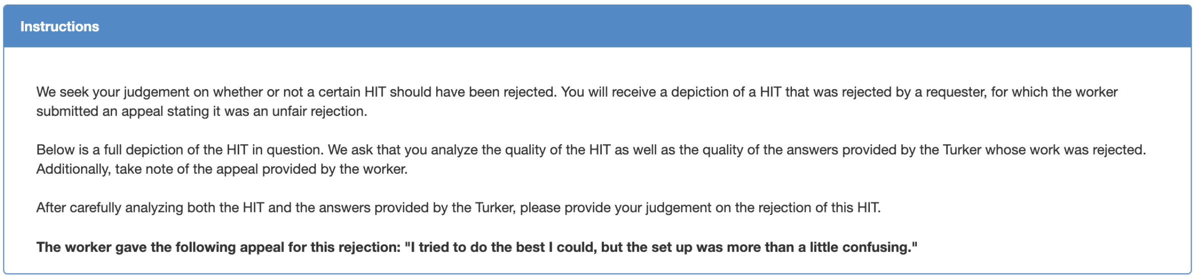
4 What the Crowd Gives You

Utilizing a crowd consisting of real individuals was necessary for the objective of our experiment. Workers were tasked with evaluating a previously completed HIT that had been rejected and deciding whether or not the rejection was warranted. While we designed our experiment and tasks to ensure that there was an objective answer to every task, our goal was to determine whether or not Mechanical Turk and other crowdsourcing platforms would be a valid and reliable way to process appeals. Additionally, the effect of appeal inclusion and economic incentives were also points to be analyzed. Therefore, due to the real-world data collection specification of our experiment, this could not have been automated.

Our experiment has broader implications for the application of crowds to task and appeal review. At present, a requester can either choose to automatically approve completed HITs (leaving open the possibility of erroneous task results for subjective tasks) or manually comb through each HIT to validate the result. Even in the latter case, having just one individual comb through thousands of data points leaves room for error. Therefore, HIT approval is inherently erroneous and, in some cases, tedious. While a worker is not negatively affected if his or her task is erroneously approved, in the opposite case, where a worker's task is erroneously rejected, workers can feel detrimental financial and mental effects. A review and appeal system as outlined in our experiment would have great implications for giving workers more power, as well as increase the validity of task results.

A system like this would be very difficult to automate because of the wide range of tasks available. This question edges into the Strong Intelligence side of Artificial Intelligence. An algorithm that could automate an appeal process would not be far off from an algorithm that could just complete tasks on Mechanical Turk, erasing its need and the need for human computation for some tasks altogether.

In this experiment, our crowd was presented with a user interface in the form of a pertinent HIT design (see the figure(a)).



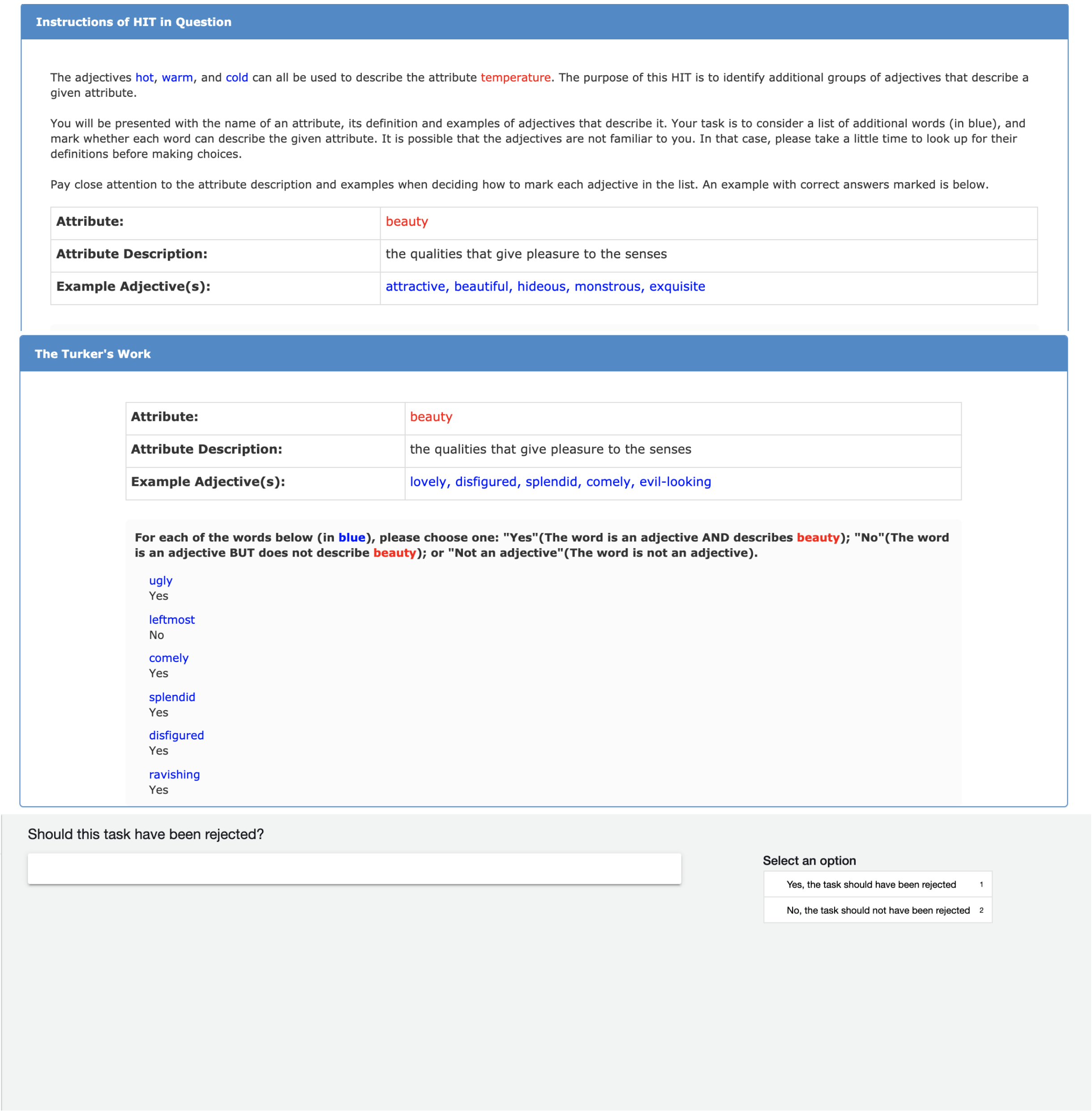
Instructions

We seek your judgement on whether or not a certain HIT should have been rejected. You will receive a depiction of a HIT that was rejected by a requester, for which the worker submitted an appeal stating it was an unfair rejection.

Below is a full depiction of the HIT in question. We ask that you analyze the quality of the HIT as well as the quality of the answers provided by the Turker whose work was rejected. Additionally, take note of the appeal provided by the worker.

After carefully analyzing both the HIT and the answers provided by the Turker, please provide your judgement on the rejection of this HIT.

The worker gave the following appeal for this rejection: "I tried to do the best I could, but the set up was more than a little confusing."



Instructions of HIT in Question

The adjectives hot, warm, and cold can all be used to describe the attribute temperature. The purpose of this HIT is to identify additional groups of adjectives that describe a given attribute.

You will be presented with the name of an attribute, its definition and examples of adjectives that describe it. Your task is to consider a list of additional words (in blue), and mark whether each word can describe the given attribute. It is possible that the adjectives are not familiar to you. In that case, please take a little time to look up for their definitions before making choices.

Pay close attention to the attribute description and examples when deciding how to mark each adjective in the list. An example with correct answers marked is below.

Attribute:	beauty
Attribute Description:	the qualities that give pleasure to the senses
Example Adjective(s):	attractive, beautiful, hideous, monstrous, exquisite

The Turker's Work

Attribute:	beauty
Attribute Description:	the qualities that give pleasure to the senses
Example Adjective(s):	lovely, disfigured, splendid, comely, evil-looking

For each of the words below (in blue), please choose one: "Yes"(The word is an adjective AND describes beauty); "No"(The word is an adjective BUT does not describe beauty); or "Not an adjective"(The word is not an adjective).

ugly
Yes

leftmost
No

comely
Yes

splendid
Yes

disfigured
Yes

ravishing
Yes

Should this task have been rejected?

Select an option

Yes, the task should have been rejected 1

No, the task should not have been rejected 2

Figure(a): a depiction of our HIT design. Note that in the HITs without an appeal, we took away the bolded line in the "Instructions" section.

5 Skills

Workers were required to have a moderate grasp on English and understand the concept of attributes and adjectives. As long as this requirement was met, workers were able to complete tasks at relatively equal proficiencies due to the more concrete nature of language. No crowd skill analysis was performed; however, we were able to increase the likelihood of skilled workers by employing several quality control measures (see section [6]: Quality Control).

6 Quality Control

Maintaining a high standard of quality control for workers was of utmost importance for our objective. Because this appeal task required reviewing the quality of other non-trivial tasks, it was important that workers 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 greater than 100
2) Location is one of Australia, Great Britain, United States

3) HIT Approval Rate for all Requesters' HITs greater than 90 percent

Criteria (1) and (3) made it more likely for workers to return higher-quality results based on their performance on other previous HITs. Criterion (2) 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 the economic constraints and our desire to analyze the economic feasibility of our system, we decided to only use the no-cost filters (see section: Economic Analysis).

Additional quality control measures beyond these presets were deemed unnecessary. This is partly because one of our main objectives was to evaluate the feasibility of an appeal system based on the judgment of workers. Therefore, placing further restrictions on workers would limit the size of the pool of eligible workers, which might pose problems when designing a related system of scale (see section [8]: Scaling Up) due to the need for multiple workers per appeal. In addition, due to the fact that our ruling system requires strict majority and for workers to come to consensus, employing quality control measures after the aforementioned judgments have been made by each worker may lead to a split judgment in some cases.

We considered the possibility of having workers do a couple gold standard attribute-adjective pairings. However, we decided not to do this because of the economic constraints of the project. In continuing this project, only considering the judgements of workers who do well on gold-standard questions opens the possibility of improving the accuracy and precision of our system.

Implicitly, our system functions without QC 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.

We did not compare the quality of task results against a gold standard due to the nature of our experimental objective. However, further avenues of related experimentation could explore the effect of various quality control standards on our results.

7 Aggregation

We performed three main analyses on our data to answer two larger scope questions. Is an automatic appeals process for Mechanical Turk fair? Is it economically efficient? For the first question, we wanted to see if there is a potential of fairness and objectivity in an appeals process. If Turkers automatically approve or reject an appeal without considering if the original Turker was right or wrong, then this process is not efficient and unfair. If the majority of Turkers approve the appeal when the Turker was objectively right, and reject the appeal when the original Turker was objectively wrong, then we have achieved efficiency and fairness in our classifications of HITs.

Next, we looked to see how the automatic appeals process is affected if the appeal from the original Turker is included in the new HITs we create. Will Turkers be more likely to side with other Turkers, and approve all appeals (whether or not the original Turker was right or wrong) if the reason for their appeal is included in the HIT? Or will this added aspect of the appeals process not affect the objectivity and fairness of the automatic appeals process? The accepted appeals vs. rejected appeals rate were compared with the gold standard labels automatically calculated to see if including an appeal caused a significant statistical change.

To calculate whether a rejected HIT was correctly or incorrectly rejected, we looked at the positive and negative quality control adjectives. There were 5 positive quality control adjectives for which the objective correct answer was "Yes," and 1 negative quality control adjective for which the answer was "No." If the rejected worker correctly answered at least 5 of these 6 quality controls, we labeled the HIT as an unfair rejection. If more than one of the quality control adjectives were given an incorrect label, we label the original HIT as a fair rejection. We refer to these as the "true labels".

Now, we must create notation for the labels output by our aggregation model. Each individual HIT was judged 5 times, meaning that for the HIT to receive a majority label of 'unfair,' atleast 3 Turker votes must label that HIT 'unfair.' If the majority voted that an appeal was rightly rejected, we labeled the majority label as fair, and if the majority voted that the rejection should be overturned, the majority label was unfair. We refer to these as the "majority labels."

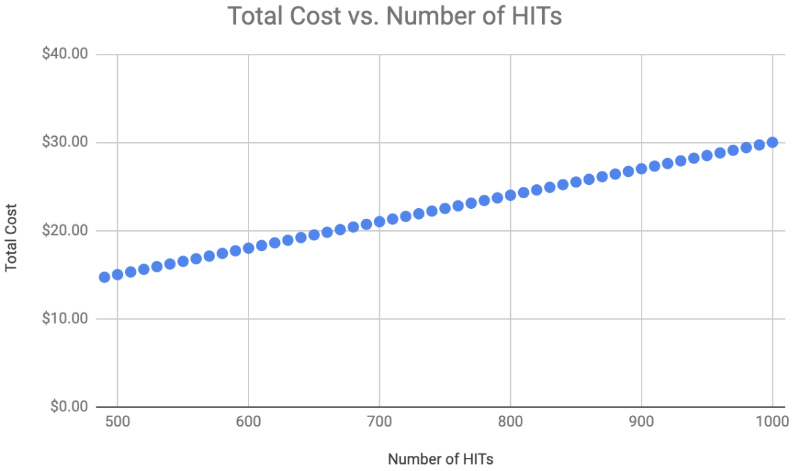
Using python scripts to aggregate, we calculated the true and majority labels for both csv files (with reasons for appeals and without reasons for appeals). Precision metrics (as specified below in section [10.1]: Precision Metrics) were then calculated from these aggregated csv files.

8 Scaling Up

In terms of scale, hundreds of thousands of tasks are completed per day across multiple task crowdsourcing platforms, including Amazon Mechanical Turk and CrowdFlower. Therefore, if our system scaled to encompass a large majority of these tasks, the amount of contributions needed would scale linearly with the amount of tasks. For the purposes of this experiment, having access to data of this size would allow for more conclusive evidence to support our hypothesis. In exchange for higher costs and computation time (both human and non-human), a larger crowd would lead to greater validity and reliability in our experimental results.

However, there exist some challenges to consider when scaling to a large crowd. On the data collection side, ensuring that a substantial amount of the workers be unique may pose a challenge. Out of our two batches consisting of 245 tasks each, 56 unique workers completed our HITs. It was important for us to get a wide variety of unique responders to judge our tasks to have a larger sample size of unique judges for our analysis. Therefore, when scaling up, it would be important to take measures to ensure this. One possible solution would be to divide the large batches into multiple batches, and publish batches iteratively as to prevent workers who worked on previous batches to keep continue to work on other batches.

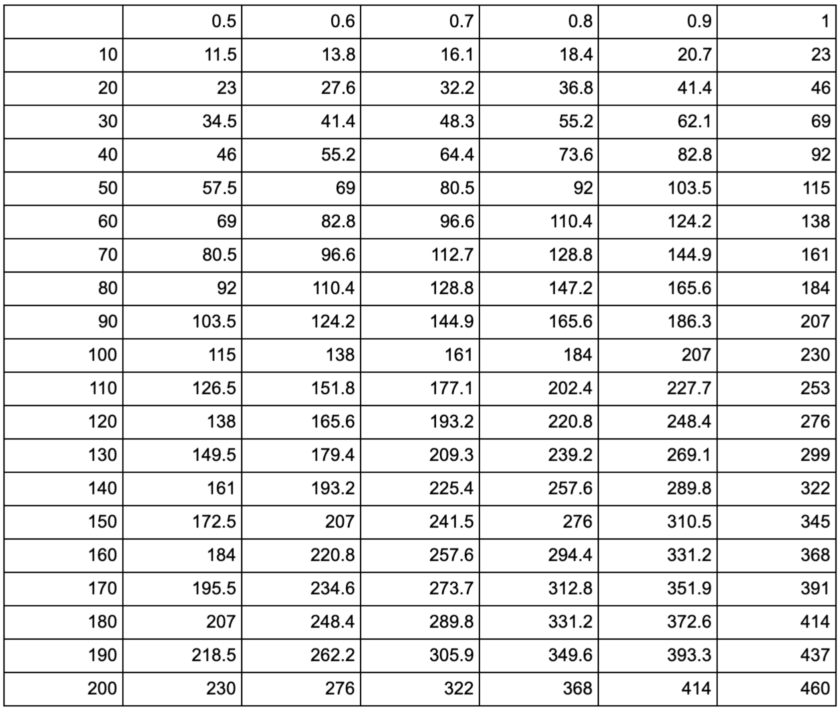
We performed a cost analysis for publishing more HITs. The batches in our experiment cost a total of $7.35 each, for an overall total of $14.70 for both batches. $2.45 of the $7.35 cost was given to Amazon as a fee ($0.01 per task). The remaining $4.90 was the payout for the 245 HITs, with a reward of $0.02 each. We looked at the cost for publishing our batch of 490 HITs total and scaled it linearly to publishing 1000 HITs. For 1000 HITs, we end up having a total cost of approximately two times the cost for publishing our original HITs.



We investigated whether this would be a cost-effective solution for requesters to accept or reject HITs 'automatically' rather than having a requester look at each individually. To investigate this, we performed an economic analysis, including on a system of larger scale (see section [9]: Economic Analysis).

9 Economic Analysis

To perform the economic analysis to determine the economic benefit of our proposed solution, we considered two different viable solutions for our product. The first is if we only charged for our service for each appeal made for a rejected HIT. In this case, it would be an automatic service for a Turker if their work was rejected, they could automatically appeal their HIT and then we would automatically post their appeal for other Turkers to reject or accept. In this case, we assumed there were 1000 HITs in the original task. In increments, we assumed anywhere from 1% of the HITs were rejected to 20% rejections. Then, we assumed anywhere between 50% of these rejections would appeal to 100% appeal rate. The number of appeals then ranged from 5 to 200 for a task of 1000 HITs. We assumed that each minute of the requestor's time was valued at 50 cents, and to look at each appeal from a rejected HIT would take 5 minutes of the requestor's time. So, the time value of looking at an appeal is 5 * .5 = $2.50. So, if there were x appeals, the requestor would spend $2.50 * x dollars in time-value to evaluate all of the appeals. If the requestor used our service, we decided to charge $0.20 per appeal for the requestor. We reached the price of $0.20 because each appeal would need to be looked at by 3 Turkers (3 * .02 cents we pay each Turker), and Amazon also takes a small fee for a premium service. We want to operate at a slight profit. However, we could minimize our price further and still make a profit, or charge just enough to break even. To calculate the monetary benefit for the requestor in terms of time-money-value, we subtracted the amount he would have to pay for our service from the time-value he would have to pay if he manually looked through each appeal. The values are shown below. The first column represents the number of rejected HITs out of the group of 1000, and the first row represents the percentage of those rejected HITs who appeal. All of the other values represent the money (in dollars) the requestor saves by using our automatic appeals process.



	0.5	0.6	0.7	0.8	0.9	1
10	11.5	13.8	16.1	18.4	20.7	23
20	23	27.6	32.2	36.8	41.4	46
30	34.5	41.4	48.3	55.2	62.1	69
40	46	55.2	64.4	73.6	82.8	92
50	57.5	69	80.5	92	103.5	115
60	69	82.8	96.6	110.4	124.2	138
70	80.5	96.6	112.7	128.8	144.9	161
80	92	110.4	128.8	147.2	165.6	184
90	103.5	124.2	144.9	165.6	186.3	207
100	115	138	161	184	207	230
110	126.5	151.8	177.1	202.4	227.7	253
120	138	165.6	193.2	220.8	248.4	276
130	149.5	179.4	209.3	239.2	269.1	299
140	161	193.2	225.4	257.6	289.8	322
150	172.5	207	241.5	276	310.5	345
160	184	220.8	257.6	294.4	331.2	368
170	195.5	234.6	273.7	312.8	351.9	391
180	207	248.4	289.8	331.2	372.6	414
190	218.5	262.2	305.9	349.6	393.3	437
200	230	276	322	368	414	460

As can be seen above, the requestor can save anywhere from $11.5 to $460 by automatically evaluating appeals instead of manually evaluating appeals. Even though these savings mostly represent time-value savings, we believe the time-value of the requestor is extremely important to him. A huge reason to turn to Mechanical Turk in the first place as a requestor is to get other people to complete your tasks that you may not have time to do.

Then, we wanted to look at what would happen if our service were implemented to automatically approve or reject every HIT to remove the middle man of the requestor manually approving or rejecting HITs. So instead of our service being an appeals process, it would pay Turkers to decide whether to approve or reject a HIT. We assumed that a requestor would spend 2 minutes on each HIT deciding to reject or approve a HIT, and we still assumed his time value of every minute to be 50 cents. So for a task of 1000 HITs, his monetary time value would be 1000 * .5 * 2 = $1000 to approve/reject all HITs. If we charge $.20 cents per HIT to approve/reject each HIT for the requestor, this would be a premium cost of $200. So, the requestor would save 1000-200 = 800 dollars in time value by using our service. There could be even higher savings depending on the salary of the requestor. For example, the average salary for a research scientist (who would be posting tasks on Mechanical Turk) is $72,707. If he works 52 weeks a year, 5 days a week, 8 hours a day, his time value per minute is 58 cents. So his monetary savings by using our service is $960.

Even though the above analysis is assumption-based, we believe there is great economic benefit from our service. If our service was implemented to approve or reject all HITs for the requestor, he would save greatly depending on his actual time/value per minute. If our service only looked at appeals for rejected HITs, the table above shows how great the time value savings could be no matter the number of rejected HITs, or the number of percentage of workers who appeal their rejected HIT. To scale this up or down, we would just have to change the 1000 number representing the number of HITs in the task to reflect the size of the new task.

10 Project Analysis

10.1 Performance Metrics

In order to analyze our results, we decided to focus our attention to the true positive rate, or recall and precision of our model. To show potential in being a productive model to resolve these rejection conflicts, we decided that our model should yield a high recall rate without precision being too low. This would mean that our workers are identifying a majority of the cases in which a rejection should have been overturned, without the workers simply overturning every rejection.

Our hypothesis was that the rate of recall of this model would be high [.8, .9] and that the precision rate would be lower than this [.5, .6].

We plan on comparing the performance metrics under the two scenarios of with and without appeal. The numbers above are [with appeal, without appeal]. This will give us a sense of whether or not appeals aid in the ability of workers to judge HIT quality.

The true labels were computed with moderately high expectations of work. Specifically, we give a label of ‘UNFAIR’ to HITs which were completed with greater that 80% quality control accuracy (in this case, 5 out of 6 quality control questions) and ‘FAIR’ to HITs which were completed with quality control accuracy less than this. Computing the true labels in this manner allows us to compare judgement quality of our workers against our moderately high expectations of HIT quality.

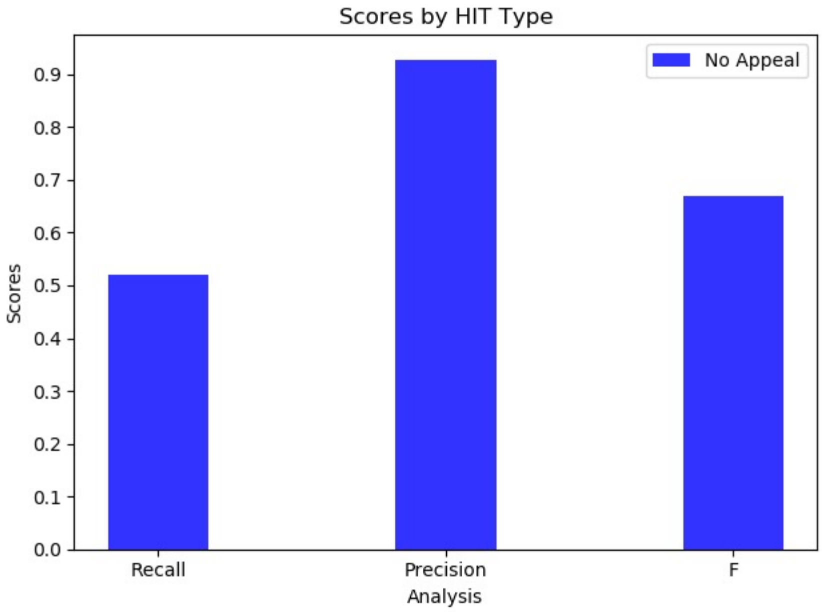
In order for our workers to show potential in being able to effectively judge HIT completion quality, we decided that our model should yield an F-score of at least .6. Because F-1 score is a metric of both recall and precision, this allows us a metric to evaluate the worker quality without directly attributing their success to recall or precision.

10.2 Results

Below are confusion matrices for our two models: first without appeals then with appeals.

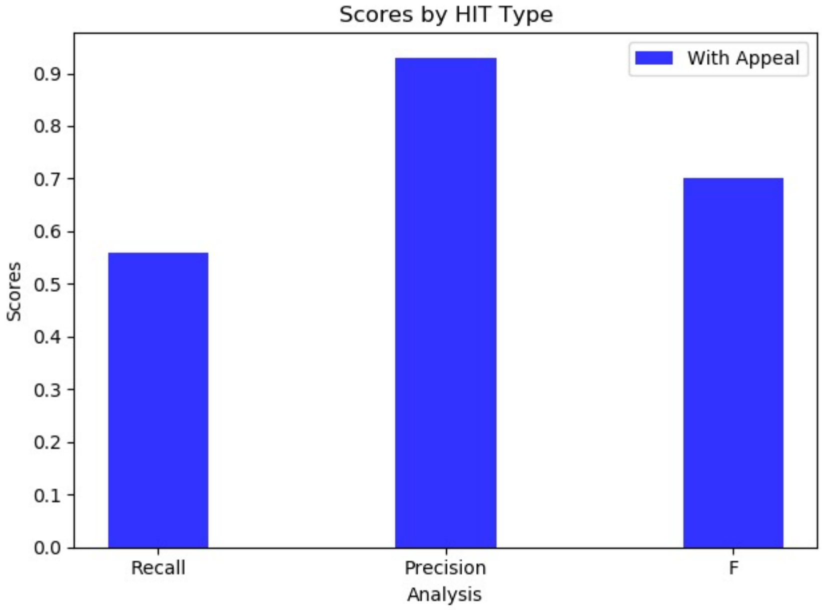
$$\begin{bmatrix} \text{Actual / Predicted} & \text{Predicted: Unfair} & \text{Predicted: Fair} \\ \text{Actual: Unfair} & TP = 14 & FN = 11 \\ \text{Actual: Fair} & FP = 1 & TN = 23 \end{bmatrix}$$

(a). Confusion matrix for HITs without written appeals



$$\left[\begin{array}{c|c|c} \text{Actual / Predicted} & \text{Predicted: Unfair} & \text{Predicted: Fair} \\ \text{Actual: Unfair} & TP = 13 & FN = 12 \\ \text{Actual: Fair} & FP = 1 & TN = 23 \end{array}\right]$$

(b). Confusion matrix for HITs with written appeals



10.3 Analysis

The particularly high rate of precision exhibited by our crowd workers shows that over 90% of the time that our workers decided a HIT should be overturned, it actually should have been overturned according to our labels. Our rate of recall, though, was lower than expected at just over 50% in both scenarios. This means that of the rejections that should have been overturned, our workers are only overturning just over 50% of.

In both scenarios, the F-1 score of our model exceeded that of our threshold for our crowd workers to prove to be able to effectively judge HIT quality. Moreover, we can see that our F-score is bottlenecked by our low recall. This result was quite surprising, given that we hypothesized a high recall and low precision. Rather than the workers excessively overturning rejections, we instead found that they were quite selective with their overturning. Our crowd workers seem to have had higher standards for overturning HITs than we did. Because the rate of recall was still non-trivial, we believe that our workers were effective in judgement of HIT quality.

The only change in judgement from without appeal to with appeal is a single true positive instance becoming false negative. Though this change may seem minimal, it is important to note that each model decision is representative of five Turker votes, making each individual model prediction more significant. We believe that this constitutes as some evidence showing that appeals cloud the judgement of our workers. Though, further work should be done for analysis of more thorough or descriptive appeals.

One of the biggest challenges we faced with was finding an objectively right or wrong task that should be relatively easy for a lot of workers to read quickly and understand to a level where they could complete it. We considered having a task where there could be a subjective question of whether it was done correctly or incorrectly, but we considered that this would be unrepresentative because our measure of whether a HIT should have been rejected or accepted is different from the unique opinion of each requester of what would be “good” work for the task.

Conclusions

From our analysis, we make 3 conclusions. First, we determined that our model proves to be economically feasible. Next, we found that workers were productive in judgement of HIT quality for a self-sustainable rejection appeals system. Turkers do not work in the interest of other Turkers when it comes to HIT appeals, and generally continue to complete work honestly. Finally, we showed that including a written appeal from the worker may cloud the judgement of the Turker who is assessing the quality of work.

11 Technical Challenges

The technical components for this experiment consisted of Python programming with specific utilization of the Pandas library for aggregation module and data analysis purposes. Amazon Mechanical Turk was another tool that was imperative for our project’s success.

One component that posed a challenge was formatting HITs within the framework of Mechanical Turk. Amazon’s system streamlines the process for some HIT designs, but for more complex HIT designs, formatting can be challenging. For example, Amazon requires specific types of div components, which makes it hard to personalize the design. Because our HIT design was important, careful consideration had to be put into designing the task in order for it to be conducive for maximum accessibility and productivity.

12 Suggestions for Further Work

Our suggestions for further work mainly lie within more comprehensive analysis of the effect of written appeals. In this paper, we have sufficient evidence to show that workers are effective in determination of HIT quality to our standards, however there is not much statistically significant data to show that appeals cloud this judgement. We believe that more descriptive appeals, as well as analyzing multiple sentiments of appeals will provide a more comprehensive review of their effect of the judgement of HIT quality.