

# Exploring Iterative and Parallel Human Computation Processes

## ABSTRACT

Mechanical Turk (MTurk) is an increasingly popular web service for paying people small rewards to do human computation tasks. Current uses of MTurk typically post independent parallel tasks. This paper explores an alternative iterative paradigm, in which workers build on each other's work. We run a series of experiments demonstrating the efficacy of this paradigm in a variety of problem domains, including image description writing, brainstorming company names, and deciphering blurry text. We also find a danger in iteration, that poor contributions can lead subsequent workers astray. More broadly, we propose a model and direction for future work in the area of automated human computation processes.

## Author Keywords

Mechanical Turk, human computation.

## ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

## INTRODUCTION

The internet is rife with examples of websites asking people to make small contributions toward a larger objective. Wikipedia is a canonical example, where well-written and informative articles grow from the contributions of many users [7].

One hurdle in these projects is motivating participants to contribute. We might try Games with a Purpose [1], but it's hard to transform everything into a game, especially if the task isn't going to be performed sufficiently many times to pay off the development cost.

Amazon's Mechanical Turk answers this question with money: it is a real-time labor market for tasks that pay on the order of cents for completion. There are people there

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2009, April 4–9, 2009, Boston, Massachusetts, USA.

Copyright 2009 ACM 978-1-60558-246-7/09/04...\$5.00.



**Figure 1: Mechanical Turk workers deciphered every word of this passage, shown nothing but the above: “Please do not touch anything in this house. Everything is very old, and very expensive, and you will probably break anything you touch.”**

every second of every day, ready to accept almost any task you give them, for surprisingly small amounts of money.

Current uses of Mechanical Turk typically center around surveys or polls. Mechanical Turk workers (or *turkers*) label images, or verify the integrity of data. Turkers are sometimes asked to write, but not in a collaborative way.

However, the variety of collaborative projects on the internet suggests that small contributions from many paid individuals are capable of achieving more. What if we could orchestrate the efforts of many workers on Mechanical Turk in creative processes, e.g.:

- Writing documents
- Creating presentations
- Brainstorming solutions
- Resolving difficult search queries
- Solving problems

Figure 1 is an example of a real challenge solved by turkers working together. This passage contains heavily blurred text, and they deciphered every word.

We are exploring an emerging model of computation, where the building blocks are tasks on Mechanical Turk. It seems appealing to be able to outsource challenging creative tasks to Mechanical Turk, but it is not yet well-understood how to orchestrate the efforts of all of these small contributions into quality creations.

The main contribution of this paper is an exploration of the possibilities, including writing descriptive paragraphs for images, brainstorming company names, and deciphering blurry text. In the process of investigating how to perform

these tasks, we break down the tasks into their core components.

The contributions of this paper are:

- Models and design patterns for automated collaborative processes that coordinate small paid contributions from many humans to achieve larger goals.
- A series of experiments that compare the efficacy of iterative and parallel design patterns in a variety of problem domains.
- An experiment that shows that paid ratings on Mechanical Turk are correlated with ratings obtained from a conventional user study.

## MODEL

We are interested in automated processes which coordinate small contributions from many humans to achieve larger goals. All of the creative and problem solving power in these processes will come from humans.

Typically this content will be in a form that a computer does not understand. In order for the automated process to make decisions, it will need to ask humans to evaluate and transform content into a form it can process. This suggests a breakdown of the domain into *generative* and *evaluative* tasks.

### Generative Tasks ■

Generative tasks solicit new content: writing, ideas, imagery, solutions. Tasks of this form tend to have few constraints on worker inputs to the system—in part because the computer doesn't understand the input. The goal of a generative task is to produce new high quality content. Many factors affect the quality of results. The mechanics of the user interface are certainly a factor.

Another factor is the instructions. In order to generate content, the worker needs to conceive of new content in their mind, and the instructions serve as constraints on their creativity. Instructions may also serve to spur or inspire creativity, by suggesting potential avenues to explore.

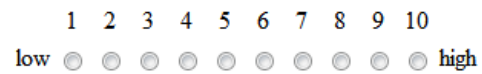
We are interested in exploring the potential benefits of showing workers content generated by previous workers. The hope is that this content will serve as inspiration for workers, and ultimately increase the quality of their results.

### Evaluative Tasks ◆

Evaluative tasks solicit opinions about existing content. Tasks of this form tend to impose greater constraints on worker contributions, since the computer will need to understand the contributions. The goal of an evaluative task is to solicit an accurate response. Toward that end, evaluative tasks may ask for multiple responses, and use the aggregate. In our experiments, we use two evaluative tasks: rating, and comparing.

### Rating

The rating task shows a worker some content, and asks them to rate the quality of the content, with respect to some goal. All rating tasks in this paper use a scale from 1 to 10, as shown:



### Comparison

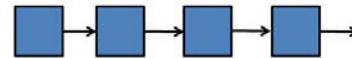
The comparison task shows a worker two items, and asks them to select the item of higher quality. The order of items is always randomized in our comparison tasks.

## Combining Tasks

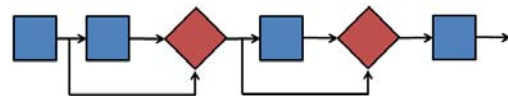
We explore automated processes that are essentially combinations of basic tasks in certain patterns. At this point, the patterns we explore are quite simple. All of the processes in this paper follow one of two patterns: *iterative* or *parallel*.

### Iterative

The iterative pattern consists of a sequence of generative tasks, where the results of each task feed into the next one. The goal is to explore the benefits of showing workers content generated by previous workers:



If it is not easy to merge the results of generative tasks, then we place a comparison task between generative tasks:

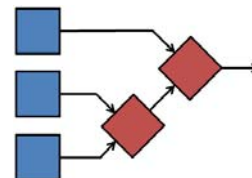


This comparison lets the computer make a decision about which content to feed into the next task. In our experiments, we want to keep track of the best item created so far, and feed that into each new generative task.

### Parallel

The parallel pattern consists of a set of generative tasks executed in parallel. The parallel pattern acts as a control in our experiments—it is effectively the same as the iterative pattern, except that no workers are shown any work created by others.

If it is not easy to merge the results, then after all the results have been generated, we employ a sequence of comparison tasks to find the best result:



## EXPERIMENTS

In this section, we describe four experiments run on Mechanical Turk. The first three experiments focus on comparing the iterative and parallel processes in different problem domains: writing, brainstorming, and problem solving. The final experiment compares subjective ratings on Mechanical Turk to ratings generated elsewhere.

### Writing Image Descriptions

This experiment compares the iterative and parallel processes in the context of writing image descriptions. Each process has six generative tasks, each paying 2 cents. Five comparison tasks are used in each process to evaluate the winning description. Each comparison task solicits five responses, where each response pays 1 cent.

The instructions for the generative tasks are shown in Figure 2. The task asks a worker to describe the image factually in at most 500 characters. A character counter is updated continuously as the user types. The “Submit” button only activates when the contents of the textarea have changed from the initial text, and there are at most 500 characters.

Note that the instruction about “using the provided text” only appears in generative tasks that have text from a previous iteration to show. These instructions are omitted in all the parallel tasks.


To compare the processes, we selected 30 images from [www.publicdomainpictures.net](http://www.publicdomainpictures.net). Images were selected based on having interesting content, i.e., something to describe. We then ran both the iterative and parallel process on each image. For half of the images, we ran the iterative process first, and for the other half, we ran the parallel process first. Turkers were not allowed to participate in both processes for a single image.

In order to compare the results from the two processes, we created a rating task. Turkers were shown an image and a description, and they were asked to rate the quality of the description on a scale of 1 to 10. We obtained 10 ratings for each image description to compute an average rating.

Our hypothesis was that the iterative process would produce better results. We reasoned that workers would be willing to spend a constant amount of time writing a description, and they could do more with that time if they had a description to start from.

### Results

Figure 2 shows an example image, along with the resulting description from both the iterative and parallel processes. If we average over the final description generated in each process for all 30 images, we get a significant difference in



- Please describe the image **factually**.
- You may use the provided text as a starting point, or delete it and start over.
- Use no more than 500 characters.

The picture is about a girl holding a yellow guitar. she is pretty and she is wearing a cap.

character count: 95/500

Submit

---

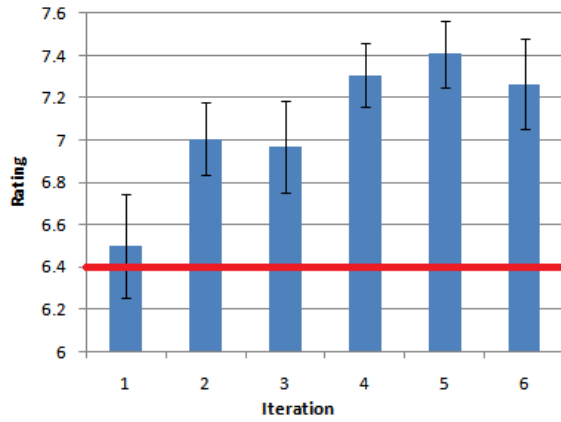
**Iterative:** A brown eyed young woman cradles an acoustic guitar in front of her chest. Her head rests against the neck of the guitar. She has long, brown hair and wears a slouchy brown hat, a gray knit long sleeved top, and two silver rings. She is either Caucasian or of Hispanic descent. Her shirt sleeves extend past her wrists. Her hands are holding the neck of the guitar. (*rated 9.1*)

---

**Parallel:** A young girl poses with her acoustic guitar. She resembles a young Sandra Bullock with dark eyes, full lips and long dark hair. She wears a newsboy-type cap a bit askew on her head, and a long-sleeve gray top; the sleeves are a bit long and extend past her wrists. (*rated 7.6*)

**Figure 2: Turkers are asked to write a factual description of an image. Turkers in the iterative condition are shown the best description so far, while the parallel task always shows an empty text area. The resulting descriptions from the iterative and parallel processes for this image are shown.**

favor of iteration (7.7 vs. 7.4, paired t-test  $T_{29} = 2.1$ ,  $p = 0.04$ ). If we average all the descriptions written within each process, the difference is still greater (7.1 vs. 6.4, two-sample t-test  $T_{358} = 5.6$ ,  $p < 0.001$ ). This suggests that giving turkers descriptions to start with helps or inspires them to write higher quality descriptions.



**Figure 3: Average rating given to descriptions written in each of the 6 iterations of the iterative processes. Red line indicates average rating of descriptions from the entire parallel process. Error bars show standard error.**

Figure 3 shows the average rating of descriptions written in each iteration of the iterative process. The red line shows the average rating of descriptions generated within each parallel process. As expected, the red line is at about the same level as the first iteration of the iterative process, since the first turker in the iterative process is not shown anything to start from. Subsequent iterations appear to grow in quality.

In sum, it appears that iteration has a positive effect on image description writing, and that the effect increases as workers are shown higher quality descriptions to start with.

### Brainstorming

This experiment compares the iterative and parallel processes in a different domain—brainstorming company names. Each process has six generative tasks, each paying 2 cents.

The instructions for these tasks are shown in Figure 4. The task asks a worker to generate five new company name ideas based on the provided company description. The “Submit” button only becomes active when there is text in each of the five input fields.

The section that lists “Names suggested so far” only exists in the iterative condition—this list contains all names suggested in all previous iterations for a given company.

We fabricated descriptions for six companies. We then ran both the iterative and parallel process on each company description. As with the previous experiment, we ran the parallel variation first for half of the companies, and the iterative first for the other half. No turkers were allowed to contribute to both the iterative and parallel process of a single company description.

In order to compare the results of these processes, we used the rating technique discussed in the previous experiment to

- **Company details:** Our company sells headphones. There are many types and styles of headphones available, useful in different circumstances, and our site helps users assess their needs, and get the pair of headphones that are right for them.
- Please supply 5 new company name ideas for this company.

**Names suggested so far:**

- Easy hearer
- Least noisy hearer
- Hearer of silence
- Silence's hearer
- Sharp hearer

<b>Iterative:</b>	7.3 : Easy on the Ears 7.1 : Easy Listening 7.1 : Music Explorer 7.1 : Right Choice Headphone 7.0 : Great Sound Headphone ...25 more...
<b>Parallel:</b>	8.3 : music brain 7.4 : Headphone House 7.0 : Headshop 6.8 : Talkie 6.4 : headphones helper ...25 more...

**Figure 4: Turkers are asked to generate 5 new company names given the company description. Turkers in the iterative condition are shown names suggested so far. The top rated names from both the iterative and parallel processes are shown.**

rate each generated company name. Again, we solicited 10 ratings for each company name, and averaged the ratings

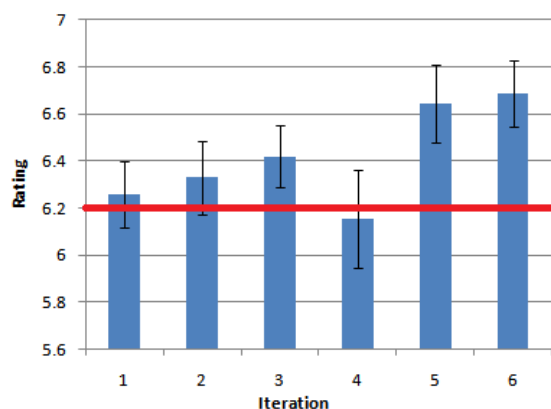
Our hypothesis was that the iterative process would produce higher quality company names, since turkers could see the names suggested by other people, and build on their ideas.

### Results

Figure 4 shows a fake company description, along with a sorted sample of the names suggested for this company. The best name generated in the parallel process is rated 8.3, compared with 7.3 for the iterative process. This difference is not significant (two-sample  $T_{18} = 1.0$ ,  $p = 0.31$ ). In fact, the parallel process generates the best rated name in 4 out of the 6 processes.

However, if we look at *all* the names generated in each process, we see a marginal significance in favor of the iterative process (6.4 vs. 6.2, two-sample  $T_{343} = 1.8$ ,  $p = 0.07$ ). This difference is more pronounced in favor of iteration if we only consider names generated in the last iteration of each iterative process (6.7 vs. 6.2, two-sample  $T_{203} = 2.3$ ,  $p = 0.02$ ). The significance of iteration becomes more clear in Figure 5, where we show the average rating of names generated in each iteration of the iterative process. The red line indicates the average rating of names in the





**Figure 5: Average rating given to names generated in each of the six iterations of the iterative brainstorming processes. Red line indicates average rating of names generated in the parallel brainstorming processes. (See the text for a discussion of iteration 4, which appears below the red line.)**

parallel process—the iterative process is close to this line in the first iteration, where turkers are not shown any names.

The average rating seems to steadily increase as turkers are shown more and more examples. The notable exception to this is iteration 4. This appears to be a coincidence—3 of the contributions in this iteration were considerably below average, each with 5 names. Two of these contributions were made by the same turker (for different companies). A number of their suggestions appear to have been marked down for being grammatically awkward: “How to Work Computer”, and “Shop Headphone”. The other turker suggested names that could be considered offensive: “the galloping coed” and “stick a fork in me”.

Overall, iteration has a positive effect on brainstorming, and showing people more suggestions seems to increase this effect.

### Blurry Text Recognition

This experiment compares the iterative and parallel processes in the problem solving domain. The task is essentially human OCR, inspired by reCAPTCHA [4]. We considered other puzzle possibilities, but were concerned that they might be too fun, which could have side effects discussed in [9]. Unlike the previous experiments, we use sixteen generation tasks in both the iterative and parallel processes, each task paying 5 cents.

Figure 6 shows an example blurry text recognition task. The instructions are simply to transcribe as many words as possible, and we place a textbox beneath each word for this purpose. In the iterative condition, these textboxes contain the most recent guess for each word.

We also ask workers to put a “\*” in front of words they are unsure about. This is meant as a cue to future workers that a word requires more attention. Note that this instruction

- Please transcribe as many words as you can.
- Put a \* in front of words you are unsure about.

**Iterative:** TV is supposed to be bad for you, but I ~~am~~ watching some TV shows. I think some TV shows are really entertaining, and I think it is good to be ~~watched~~. (94% correct)

**Parallel:** TV is supposed to be bad for you, but I like watching some TV shows. I think some TV shows are really ~~advertising~~, and I think it is good to be entertained. (97% correct)

**Figure 6: Turkers are shown a passage of blurry text with a textbox beneath each word. Turkers in the iterative condition are shown guesses made for each word from previous turkers. The resulting transcription from each process is shown, with incorrect words marked with strikeouts.**

appears in the parallel tasks as well, even though no workers see any other workers’ stars.

We also don’t need to pay any workers to rate the results, since we can assess the accuracy of the results automatically.

We composed 12 original passages. It was important to use original text, rather than text obtained from the web, since turkers could conceivably find those passages by searching with some of the keywords they had deciphered.

We then ran each passage through an image filter. The filter works on a pixel level. Each pixel in the new image is created by taking a pixel near that location in the old image, according to a 2-dimensional Gaussian. This is identical to the lossy “blur” tool from some popular image editing programs. The result is blurry text that is very difficult to decipher. Some words appear to be entirely illegible on their own. The hope is that by seeing the entire passage in context, turkers will be able to work out the words.

We applied both the iterative and parallel process to each passage. As before, each process was run first half the time, and no turkers were allowed to participate in both processes for a single passage.

The final transcription of the blurry text is extracted from each process in the following way: we look at all the guesses for each word, and take the word with the plurality of guesses. If there is a tie, then we choose one of the top contenders randomly.

Our hypothesis was that the iterative process would have a high probability of deciphering each passage, since turkers would be able to use other people's guesses as context for their own guesses. The analogy would be solving a crossword puzzle that other people have already started working on.

### Results

Figure 6 shows a passage, along with the transcription extracted from both the iterative and parallel processes. In this case, the parallel process does a slightly better job (97% of words transcribed correct vs. 94%). When we average over all 12 runs, we fail to see a statistically significant difference between each process (iterative 65% vs. parallel 62%, paired t-test  $T_{11} = 0.4$ ,  $p = 0.73$ ).

If we look at the accuracy of each process after  $n$  iterations in Figure 7, we see that both processes gain accuracy over the first eight iterations, and then seem to level off. Note that the iterative process appears to be above the parallel process pretty consistently after the fourth iteration, but this difference is never significant (even at iteration 13, paired t-test  $T_{11} = 1.4$ ,  $p = 0.19$ ).

The results suggest that iteration may be helpful for this task. However, it is also worth noting that iteration sometimes appears to get stuck due to poor guesses early in the process. For instance, one iterative process ended up with 30% accuracy after sixteen iterations. The final result was very similar to the eighth iteration, where most of the words had guesses, and they made a kind of sense:

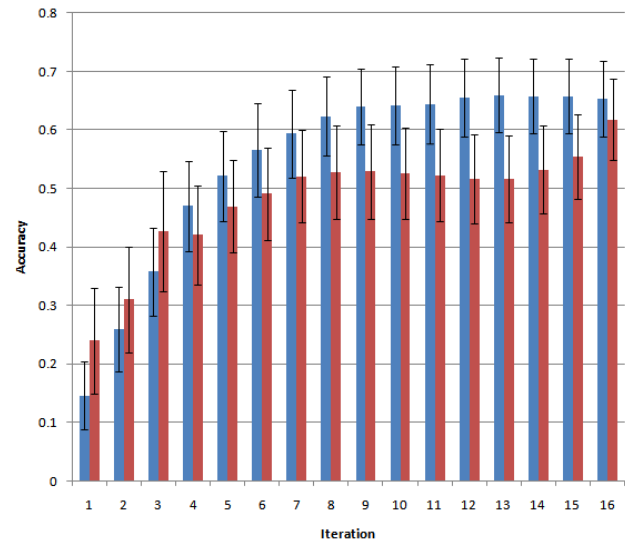
8<sup>th</sup> iteration: "Please do ask \* anything you need \* me. Everything is going fine, there \* \* , show me then \* \* anything you desire."

16<sup>th</sup> iteration: "Please do ask \* about anything you need \* me. Everything is going fine, there \* were \* , show me then \* bring \* anything you desire."

Incorrect guesses have been crossed out. Here is the actual passage: "Please do not touch anything in this house. Everything is very old, and very expensive, and you will probably break anything you touch."

Note that a single turker deciphered this entire passage correctly in the parallel process, suggesting that progress was hampered by poor guesses rather than by unreadable text.

Overall, iteration seems to have a positive effect in this domain, but we do not have enough power in our experiment to say for sure.



**Figure 7: Blue bars show the accuracy after  $n$  iterations of the iterative text recognition process. Red bars show accuracy for the parallel process after  $n$  submissions. Error bars show standard error. The overlap suggests that the two process types are not statistically different from each other.**

### Subjective Photo Rating

This final experiment is of a different kind, and compares subjective evaluations solicited from Mechanical Turk with subjective evaluations obtained elsewhere. We wanted to compare evaluations made with some domain expertise, so we chose photos, since we have ready access to many people in our department who are at least amateur photographers.

We began the study by selecting 30 random photos from [www.publicdomainpictures.net](http://www.publicdomainpictures.net). This site keeps a rating for each photo, and we ensured a uniform distribution of ratings among the 30 photos. (Note that these photos were not used in the photo description experiment.)

From the 30 photos, we created 20 tasks: 10 rating tasks and 10 comparison tasks. Each photo appeared in exactly one task. Ratings were done on a scale of 1 to 10, and comparisons were made between 2 images

We solicited amateur photographers to complete these tasks using a department mailing list, with the offer of \$25 to one randomly selected participant. We had 52 participants complete all 20 questions.

The same 20 tasks were posted to Mechanical Turk, along with additional tasks to rate the rest of the photos, and involve each photo in a comparison task. We solicited 10 turkers for each task. Turkers were disallowed from completing multiple tasks involving the same photo.

### Results

Figure 8 shows each of the 10 images in the rating tasks given to our experts, along with the average turker rating,

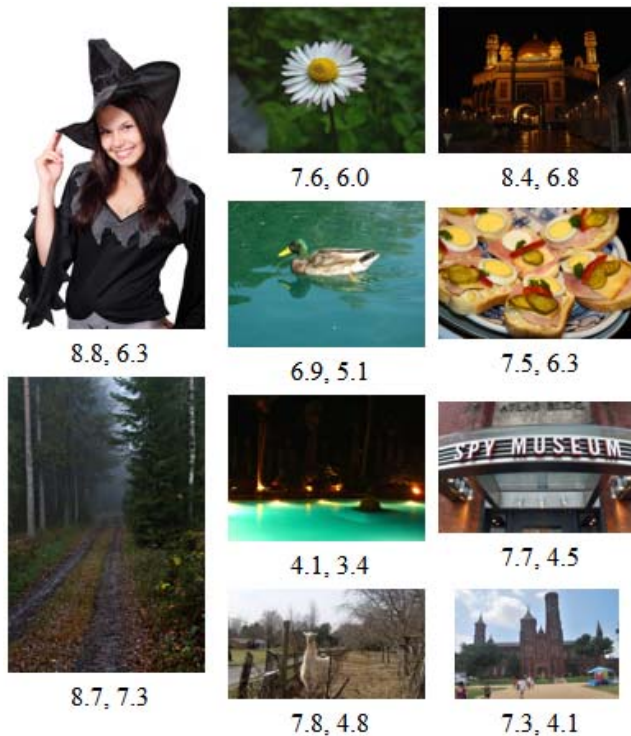


Figure 8: Each image has 2 ratings: average turker rating, and average expert rating.

and average expert rating. Figure 9 shows this information in a scatter plot: each photo is plotted according to its rating from turkers and experts. A linear regression between turker ratings and expert ratings reveals a correlation of  $R^2 = 0.6$ .

When it comes to comparing two images, we see a correlation between the images chosen by turkers, and the images chosen by experts. For every pair of images, we can plot the percent of turkers favoring the first image on the x-axis, and the percent of experts favoring the first image on the y-axis (see Figure 10). If we do this, we get a moderate correlation ( $R^2 = 0.51$ ).

To combine ratings and comparisons, we look at the number of times the average comparison between images  $a$  and  $b$  agrees with the average ratings for  $a$  and  $b$ . Turker comparisons agree with turker ratings 73% of the time, over 15 samples, which is marginally significant ( $\chi^2(1, N = 15) = 3.27, p = 0.071$ ). To see a significant relationship between ratings and comparisons, we look back at the image description experiment. This experiment conducts 300 comparisons between items, which are subsequently rated. In this case, 69% of the comparisons agree with the ratings ( $\chi^2(1, N = 300) = 44.9, p < 0.001$ ).

## DISCUSSION AND FUTURE WORK

All of these experiments demonstrate some measure of success in performing several relatively high-level creative

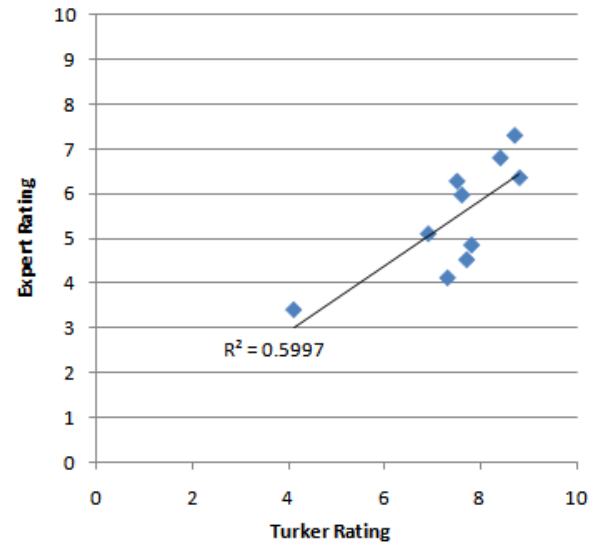


Figure 9: Each image is plotted according to its turker and expert ratings. A linear regression line is shown, along with  $R^2$ , indicating the correlation between the rating sources.

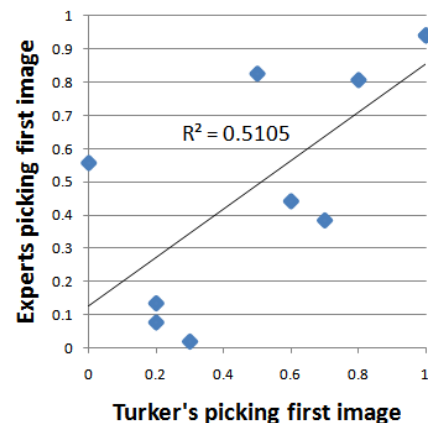


Figure 10: Each pair of images compared by turkers and experts is plotted. X-axis indicates the portion of turkers picking the first image from the pair. Y-axis indicates the portion of experts picking the first image.

and problem solving processes, using processes that orchestrate the efforts of multiple workers. We also see that the breakdown into generative and evaluative tasks is applicable to a diverse set of problem domains.

## Take Away Points

1. It does appear useful to show prior work in small creative generative tasks. In three relatively different instances, we see the same phenomenon: showing prior work positively affects output quality. It may be that we can take this idea further. In the case of writing image descriptions, it may be useful to show a different kind of prior work, perhaps a brainstorm of objects or themes worth mentioning in a

description of the image. Similarly in the brainstorming case, it might be more useful to show workers words that other people have generated—words that have to do with the company—which can be used as building blocks in names.

2. Showing prior work can have a negative effect on quality by leading future workers down the wrong path. This effect is most pronounced in the blurry text recognition task, and may be an issue in other tasks of this form where puzzle elements build on each other (like words in a cross-word puzzle). Turkers take suggestions from previous turkers, and try to make the best guesses they can, but backtracking is more rare. This may need to be programmed into the process. For instance, it may be useful to run multiple iterative processes in parallel.

3. Subjective assessments measured on Mechanical Turk seem correlated with assessments made outside of Mechanical Turk, in at least one instance. We need to explore this more, but the idea is comforting, since it suggests an efficient way of scientifically evaluating new human computation processes, in order to optimize the robustness and efficiency of algorithms that are essentially subjective in nature.

### **Future Work**

There are many directions for future work exploring automated human computation processes on Mechanical Turk. First, it seems fruitful to further explore the two basic building blocks: generative tasks, and evaluative tasks. For instance, there are many factors that may influence generative tasks, including price, how much work is expected, whether examples are shown, and whether prior work is shown. We have only scratched the surface of exploring these possibilities, but as these different elements are better understood, it will be easier to create generation steps with optimal probability of high quality and/or quantity of generated results.

Evaluative tasks also leave room for investigation. The basic goal in our tasks has usually been to determine the best items in a set, but there are a number of ways to achieve this, including absolute ratings, pair-wise comparisons, and sorting multiple items in a single task. There are even combinations of these elements that may be useful, like having people sort items, but also provide ratings. Again, this is a large space, with the promise of providing useful knowledge to help optimize evaluation of subjective content.

Along these lines, it may be useful to investigate Mechanical Turk in more basic ways, as a programming platform. For instance, it would be useful to know how factors like price, task difficulty, number of tasks posted together, time of day, (etc...), affect time-to-completion, and result quality.

Another direction for future work is exploring new building blocks. Even inside the paradigm of generative and

evaluative tasks, there is room for building blocks which sit somewhere in-between. For instance, a generative writing task could ask a turker to select which of two previous versions they want to start from (where they are effectively voting for which of those is best). Note that this could have the side effect of turkers selecting the worse version from which to start, since it may be easier to improve. Hence, these ideas need to be tested and validated.

Finally, the real creative potential in this space is exploring new high-level processes for coordinating workers to perform better, or achieve loftier objectives. For instance, in paragraph writing, one can imagine breaking down the task of paragraph writing into two steps. The first step might have people brainstorm phrases or concepts that should be included in a paragraph, and then these could be shown to the people writing the paragraph.

One might also imagine writing something larger than a paragraph through several steps: writing an outline, having separate processes to write each paragraph from the outline, and have another process to combine these results with transition sentences into a complete essay.

Another large objective is clustering: coordinating human computation to come up with categories, features and clusters of data items. These clusters have the interesting potential to have human-intelligible names or meanings assigned to them, which is a common frustration with machine learning techniques.

The ultimate goal is to learn how write processes on Mechanical Turk that reliably and efficiently achieve their objectives, and to push the boundaries of those objectives.

### **RELATED WORK**

One challenge in writing human computation algorithms is motivating humans to do work. One approach is Games With a Purpose [1] [2] [3], where humans perform useful computation as a byproduct of playing computer games. User-generated content websites such as Wikipedia use human computation to generate content, and this content along with social factors seem to motivate future contributions. Bryant [5] makes observations about how people begin contributing to Wikipedia, and what tools expert contributors use to manage and coordinate their work. MTurk provides a platform for performing Human Intelligence Tasks (HITs) where humans are motivated by money. This platform has been adopted for a variety of uses, both in industry and academia. Kittur [6] discusses how to run user studies on MTurk, while Sorokin [8] uses MTurk to label images. Thus far, the typical usage pattern for MTurk involves generating all the HITs that need to be completed, posting them to MTurk, and later downloading all the results. Several websites focus on managing HITs that fit this template (e.g. HIT-builder1). It is currently rare, however, to automatically generate new HITs based on the results of previous HITs. Sorokin proposes creating voting



HITs in response to labeling HITs, but does not report any experiments using this technique.

## CONCLUSION

This paper breaks down automated human computation processes into generative and evaluative components. We demonstrate the effectiveness of a technique in generative tasks in a variety of problem domains. We also show evidence that evaluative tasks on Mechanical Turk are correlated with evaluative tasks performed elsewhere. These results act as a foundation upon which to build and test higher-level human computation processes.

## ACKNOWLEDGMENTS

Omitted for blind review.

## REFERENCES

1. Luis von Ahn. Games With A Purpose. IEEE Computer Magazine, June 2006. Pages 96-98.
2. Luis von Ahn and Laura Dabbish. Labeling Images with a Computer Game. ACM Conference on Human Factors in Computing Systems, CHI 2004. Pages 319-326.
3. Luis von Ahn, Shiry Ginosar, Mihir Kedia and Manuel Blum. Improving Accessibility of the Web with a Computer Game. ACM Conference on Human Factors in Computing Systems, CHI Notes 2006. pp 79-82.
4. Luis von Ahn, Ben Maurer, Colin McMillen, David Abraham and Manuel Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. Science, September 12, 2008. pp 1465-1468.
5. Susan L. Bryant, et al. Becoming Wikipedian: transformation of participation in a collaborative online encyclopedia. GROUP 2005.
6. Kittur, A., Chi, E. H., and Suh, B. 2008. Crowdsourcing user studies with MTurk. CHI 2008.
7. Kittur, A. and Kraut, R. E. 2008. Harnessing the wisdom of crowds in wikipedia: quality through coordination. CSCW '08. ACM, New York, NY, 37-46
8. Sorokin, A. and D. Forsyth, "Utility data annotation with Amazon MTurk," Computer Vision and Pattern Recognition Workshops, Jan 2008.
9. Winter Mason, Duncan J. Watts. Financial Incentives and the "Performance of Crowds". HCOMP 2009.