Worker Collaboration in Crowdsourcing Markets

Jeffrey M. Rzeszotarski
Human Computer Interaction Institute
Carnegie Mellon University
jeffrz@cs.cmu.edu

ABSTRACT
Crowdsourcing platforms provide an incredible environment for studying the future of work and a pool for conducting research studies. Yet, crowdsourcing markets may not make full use of the capabilities of workers. In particular, crowdworkers might be enabled to collaborate to complete tasks, choosing and completing small portions of a large, complex task. Even further, crowdworkers might benefit from the interaction of collaboration, and may even learn from the tasks themselves. By exploring new ways to create tasks and designing next generation crowdsourcing markets, one might be able to more fully utilize the potential of crowdworkers, both for the requester and worker's benefit.

Author Keywords
Crowdsourcing, Mechanical Turk, computer supported collaborative work, content generation

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

INTRODUCTION
Crowdsourcing platforms that use a market economy, such as Amazon's Mechanical Turk, provide a tremendous environment for studying the future of human employment and economies as well as conducting research studies in fields such as HCI. Within moments, one can reach participants and receive results with a reasonable degree of accuracy and reliability. This quick, inexpensive turnaround allows for iterative study design and rapid prototyping. Even further, the participants in markets such as Mechanical Turk, Internet-savvy individuals of mixed age and gender across the globe, seem almost tailor-made for many HCI studies [3]. The low investment coupled with quality results has enabled the large-scale collection of data that might otherwise be unwieldy to gather, such as language processing corpora or user studies [2, 1].

Yet, the very features that make Mechanical Turk and other crowdsourcing markets so appealing to researchers can serve to dehumanize workers, and even limit the variety of possible tasks. Amazon's Mechanical Turk interface is designed to maintain a degree of separation between requester and worker. Workers are identified only by number, and the interaction between requester and worker is limited to HIT (Human Intelligence Task) postings, a binary approval process, assignation of qualifications, and a multi-step bonus award process. Workers themselves have no awareness of other workers on the same HIT except for the steady decrease of the count of available tasks.

This design, while effective in enabling tasks to be posted and completed, neglects one key element of crowdsourcing: highly capable humans are completing the tasks. The Turkers (workers on Mechanical Turk) are individuals able not only to follow directions to complete a task, but also infer, think creatively, socially interact, and make complicated judgments. They are not solely input/output functions that take a HIT and produce a unitary product that is either approved or denied for a given payment. Crowdsourcing markets such as Mechanical Turk largely feature task construction conventions and reward systems that are structured around this abstracted unitary task completion and response paradigm, perhaps at the expense of other possible interfaces and interactions. Because of this limitation, it is worth exploring what methods might enable workers to leverage even more of their potential.

In particular, humans are well adapted towards working in groups and organizing labor. Workers in their daily life may be assigned to complete a part of a large project in their occupation. Why, then, must workers in the crowd online always work alone with the requester combining their small contributions together? There might be untold gains in allowing crowdworkers pool resources and collaborate to complete larger work projects.

This leads to two fundamental questions: How can pre-existing markets like Mechanical Turk be finessed into supporting more complicated and *human* tasks such as collaboration? What decisions might go into the design of a new market that helps participants collaborate and pool resources?

ENHANCING MECHANICAL TURK HITS
Many Mechanical Turk HITs take the form of a single, unitary task that produces a single, unitary result. These individual result units, be they tags for an image or sentence translations for machine translation training, are later curated into a greater product by the requester. The Turkers have a limited view of the final project in mind, and work largely without an understanding of the larger context of the HIT. However, as humans, Turkers are capable of interpreting intention and placing their work in a larger body (as evident in other crowd efforts, such as Wikipedia).

To enable Turkers to produce a complete product rather than a part of the whole while at the same time harnessing the crowd rather than an individual, I generated a series of collaborative Turker tasks investigating a variety of approaches. Turkers are instructed in the HITs to proceed to a collaborative text editor and work together, interacting socially, to complete a text generation task. The task was declared as completed by group consensus, and individual Turkers were compensated based on completing a *portion* of the task, rather than the entire task.

Collaborative Translation

I examined translation in a collaborative context in a series of HITs. Turkers were tested on their Spanish language vocabulary then asked to work together to translate an English text into Spanish. They logged into a persistent collaborative text editing environment, EtherPad (Figure 1), through the HIT and contributed a portion of the final translation. Credit was given for translating several sentences, and social interaction was fostered through an instant chat interface.

I solicited 88 participants in total for 3 different translation passages of varying content, each divided into three different collaborative work groups (9 in total). After 4 days, I was able to post-test 49 Turkers, asking both about vocabulary and the contents of the passage. Concurrently, I requested 15 independent translation HITs, 5 per passage, for comparison. 4 individual translations were excluded for being identical to Google translations, and one collaborative translation was excluded as only 1 user logged in. Several effects emerged out of the individual and collaborative translations. The collaborative translations, rated by another set of Turkers, were not significantly different in quality from individual translations, although passage content played a role with one difficult passage having marginally more errors when translated by a group ($F(5,1)=2.85$, $p=0.06$). This suggests that there is indeed some cost to collaboration depending on the content of what is being translated. However, each individual worker in the group case spent far less time working than the individual translators who had to translate the entire passage.

I also examined learning in the translation task. While it is obvious that the worker benefits from a task monetarily, other benefits to the worker have largely not been explored. In the case of translation, the worker might recall or learn from the passage he or she is translating, or perhaps might learn a new vocabulary word after asking in chat. Indeed, for two of three passages, approximately half of the chat in the text editor related to vocabulary questions and clarifications. While no differences between pre vs. post vocabulary test questions achieved significance, possibly because group translators did not translate portions that included tested vocabulary words, all translators, individual and collaborative alike, showed recall of the contents of what they translated.

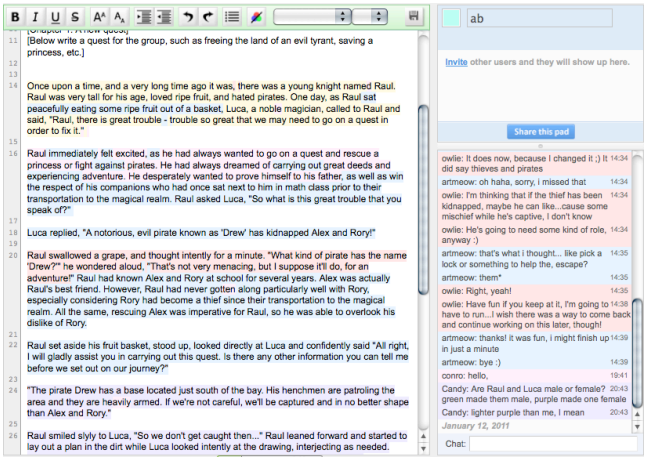

Figure 1: EtherPad collaborative text editing interface

Two raters rated post-test summaries of what Turkers thought they had translated on scales from 1-5 for both accuracy and completeness, with 5 being perfect. One passage demonstrated a combined mean of 4.20 accuracy and 4.08 completeness, showing that translators recalled the passage in a high degree. A more difficult passage was recalled less, with 3.54 accuracy and 2.67 completeness. Even then, Turkers recalled some of the meaning of the text they had translated. A two-way ANOVA revealed that passage contents were closely linked Turker recall ($p=.06$ for accuracy, $p=.05$ for completeness).

Collaborative Turkers overall produced products roughly equal in quality to independent Turkers, although individually translating far fewer sentences. Further, both independent and grouped Turkers remembered a fair amount about the passage they translated 4 days later. This presents the interesting possibility of creating HITs that also inform or educate Turkers while they work.

The collaborative, persistent nature of the translation task also served to retard cheating behaviors. In environments where many users were contributing, there was little cheating behavior. However, several individually made translations were clearly copied from machine translations. Perhaps the social pressure of a collaborative environment encoded some level of accountability in the task. Further encouragement could be introduced by rewarding all collaborative workers for producing a quality final product.

Collaborative Story Writing

The positive results in HITs involving Turkers contributing as a group towards constructing a complete, quality translation without requester intervention suggest that even more sophisticated tasks might be achievable on crowdsourcing markets. I have examined purely creative, content generation tasks, namely writing stories for educational software. Writing narratives to make educational software such as cognitive tutors more interesting is an expensive task, and crowdsourcing seems like an excellent avenue to produce quality, cheap stories.

As before, Turkers were directed towards a persistent collaborative text editor. The HIT described the general purpose, the creation of a story for sixth graders, and provided a general narrative framework (a barebones list of characters and setting). From there, as before, Turkers were required to contribute a few sentences that fit within the story plot and were grammatically correct.

This task is by nature completely open-ended. Turkers successfully wrote stories of varying quality. In all cases, the stories required revision and did not have a conclusion. Unlike in translation, the story has not discrete end state and can continue possibly ad infinitum. As the story evolved, Turkers required context-sensitive directions which were difficult to implement in the default Mechanical Turk interface. However, I simulated this by splitting the story-writing task into multiple passes, where workers first work together to build an outline, then another set writes the story, then another revises and makes sure it matches a middle school reading level and so forth.

The open-ended, creative story writing demonstrated the costs of collaborative work more clearly than fixed-length translations. As the story grew, the characters and plot drifted from its initial framing, in several cases becoming nonsensical. After many Turkers contributed, the story was long enough that future contributors may not want to read and understand the previous work in order to make a worthwhile contribution. As more Turkers worked, this cost only became higher. These costs are not unique to Turkers, but manifest in many collaborative environments.

In each story HIT, the Turkers demonstrated consensus-building behaviors in the chat function, talking about character motivations, setting consistency, and ideas for future directions before they left the environment. The Turker comments submitted after leaving the text editor were largely positive, reflecting a high degree of interest in writing future stories, seeing the finished product after more Turkers worked, and refining the story again on their own.

Future Mechanical Turk Goals

While the Mechanical Turk system is not absolutely flexible, the combination of third party sites and the integrated HIT API as demonstrated above empower Turkers to apply more of their inherent potential in tasks. Turkers were not only able to make a multiple sentence contribution as they would have done in a normal HIT, but also revise the work of other Turkers, participate in a social group centered around a task, and gain a more complete understanding of how their contributions were going to be applied. These collaborative proof of concepts are just a tiny subset of possible Turker-curated work. One can imagine multiple Turkers working together to complete and integrate units of larger tasks, such as product buying guides, blog post writing, citation checking, transcribing and summarizing lengthy documents, and reliably coding large datasets, or even managing each others' work.

The richness of the API can further enable dynamic HITs that can, among other things, resolve some of the problems inherent in the story-creation HITs. As more workers contribute, the instructions might mutate, enforcing more revision roles as a text matures. In the final stages, Turkers might even be able to vote on whether a product seems finished, or if in their judgment it needs another cycle of work.

SOCIAL CROWDSOURCING

While Mechanical Turk can be made to go a long way towards using more of the abilities of workers, its strict anonymity and worker/requester disconnect does not allow for more social interaction among Turkers, which might enable a host of collaborative and worker-generated tasks. Already Turkers gather in third party forums to discuss good and bad requesters (given the asymmetry of approval ratings on the site) and HITs they enjoy. A future crowdsourcing market might incorporate this social interaction directly into its design.

This new, social crowdsourcing market would be focused around temporary collaborations between crowd-workers, taking small portions of a larger problem and working together in an ad-hoc fashion to build a successful product. In this market, HITs might take the form of persistent environments that workers can enter and receive payment based on the size and importance of their contribution rather than single unit tasks that contribute to an unclear greater goal.

Because of the collaborative nature of such tasks, social interaction and motivation are critical. As a result, one might examine the influence of social motivation techniques including social networking, group membership, and reputation systems on products and worker performance. Further, cheating might manifest differently in such markets, as accountability is now placed not only in the hands of the requester, but also concerned workers who might contribute by weeding out cheating responses. Finally, compensation must be re-evaluated given the graduated nature of contributions and increased social interaction about tasks between workers and requesters.

CONCLUSION

Crowdsourcing markets provide a huge arena for studying economic, employment, and social effects, along with a pool for study participants. However, the conventions and designs of current crowdsourcing markets do not necessarily take full advantage of what workers may be capable. In particular, crowdsourcing market design, reward structures, and task conventions are not well tailored for collaborative work, which in the case of translation has been shown to produce quality work with less time spent by each individual worker. Further, this sort of work allows workers themselves to create a final product rather than requiring the requester to curate many small submissions, and allows for tasks that may not have a definite end state

such as creative writing. By pushing the boundaries of task construction, one can explore these new avenues and better use the capabilities of workers, both for the benefit of the requester and the benefit of the worker themselves.

SHORT BIOGRAPHY

Jeff Rzeszotarski (rez-oh-tar-ski) is a first year Ph.D. student in the Human Computer Interaction Institute at Carnegie Mellon University interested in crowdsourcing and social computing. Before joining the HCII, Jeff graduated from Carleton College with a bachelor's degree in computer science magna cum laude with distinction in major. His senior thesis project concerned computer-supported therapy for Alzheimer's Disease patients using graph-based dynamically generated presentations of the patient's life. Jeff is advised by Niki Kittur, and is currently researching ways to condense Internet resources, such as Wikipedia page histories, to help both new and current users build a general understanding of content. He also is interested in online social interaction, the HCI applications of crowdsourcing, and the behavior of crowdsourcing workers.

REFERENCES

1. Callison-Burch, C. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. *In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing,* Association for Computational Linguistics, *(2009),* 286–295.
2. Kittur, A., Chi, E., and Suh, B. Crowdsourcing user studies with mechanical Turk. *Proceedings of the twenty-sixth annual SIGCHI conference on Human Factors in Computing Systems*, (2008), 1509-1512.
3. Ross, J., Irani, L., Silberman, M., Zaldivar, A., and Tomlinson, B. Who are the crowdworkers?: shifting demographics in mechanical turk. *Proceedings of the 28th annual SIGCHI international conference extended abstracts on Human factors in computing systems*, ACM (2010), 2863–2872.