The Effects of Sequence and Delay on Crowd Work

Walter S. Lasecki[1], **Jeffrey M. Rzeszotarski**[2], **Adam Marcus**[4], **Jeffrey P. Bigham**[2,3]

Computer Science Department[1]
University of Rochester
wlasecki@cs.rochester.edu

HCI[2] and LT[3] Institutes
Carnegie Mellon University
{jeffrz,jbigham}@cs.cmu.edu

Locu / GoDaddy[4]
marcua@marcua.net

ABSTRACT

A common approach in crowdsourcing is to break large tasks into small microtasks so that they can be parallelized across many crowd workers and so that redundant work can be more easily compared for quality control. In practice, this can result in the microtasks being presented out of their natural order and often introduces delays between individual microtasks. In this paper, we demonstrate in a study of 338 crowd workers that non-sequential microtasks and the introduction of delays significantly decreases worker performance. We show that interruptions where a large delay occurs between two related tasks can cause up to a 102% slowdown in completion time, and interruptions where workers are asked to perform different tasks in sequence can slow down completion time by 57%. We conclude with a set of design guidelines to improve both worker performance and realized pay, and instructions for implementing these changes in existing interfaces for crowd work.

Author Keywords

Crowdsourcing; human computation; workflows; continuity; interruptions; efficiency

ACM Classification Keywords

H.4.2 Information Interfaces & Presentation: User Interfaces

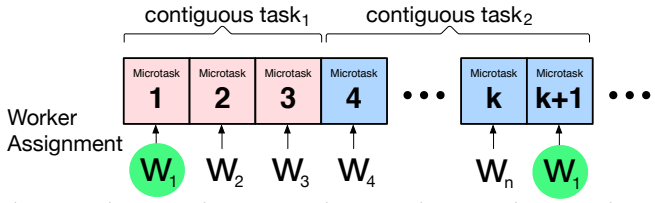
INTRODUCTION

Creating crowdsourcing workflows often involves breaking down large tasks into microtasks that can be parallelized across many crowd workers and that are amenable to redundancy-based quality control. In practice, this often results in temporal and contextual interruptions for workers that can reduce their efficiency when they want to work for longer on a single task. This is because workers must spend time recovering their working context after each change or delay.

In this paper, we identify two types of interruptions that are harmful to worker efficiency and quantify their costs: *(i)* contextual interruptions, in which workers swap between tasks of different types, and *(ii)* temporal interruptions, in which workers must wait between submitting one task and receiving the next one. Contextual interruptions can arise as a result of workflows that do not present related tasks in sequence. For example, a transcription task might ask workers to transcribe audio from different parts of a clip or different clips instead of allowing them to build on context by transcribing sequeuntial clips. Temporal interruptions can be caused by platforms that add procedural delays between tasks, *e.g.*, loading screens, steps to move to a new task, or by not using prefetching.

At first, it seems counterintuitive that task designers would present work out of sequence when a natural ordering exists, but this is quite common given that platforms commonly route workers to tasks with the least number of existing completed labels rather than sorting and delivering tasks in the order in which they were created. When requesters test the system themselves, they see the tasks presented in sequential order because they are the only ones accessing them. In practice, when many tasks are posted to a platform like Mechanical Turk, workers start at nearly the same time, so the default ordering leads workers to receive tasks out of order (Figure 1). For example, if there are n workers completing a large set of microtasks at the same rate, then the next task that each worker is asked to do will be $n-1$ tasks away from the last one they completed as no one has completed them yet. For example, if 20 workers are completing microtasks that involve transcribing 30 seconds of audio (common on platforms like Mechanical Turk), each new segment of audio received will be $30 * 19 = 570$ seconds from the previous, eliminating the context that sequential segments would provide.

In this paper, we supplement the existing literature on worker preferences with a survey of 100 Mechanical Turk workers. We find that contextual interruptions commonly result from workers switching between unrelated tasks. We then test the impact of temporal and contextual interruptions on workers



id	labels	image	inside	# of ppl	# of trees	...
...	...	...	...	...	...	...
2345	2	house00078.jpg	TRUE	1	0	...
2346	2	house00079.jpg	TRUE	1	0	...
2347	2	house00080.jpg	TRUE	1	0	...
2348	1	park0001.jpg	FALSE	0	12	...
2349	1	park0002.jpg	FALSE	0	14	...
2350	1	park0003.jpg	FALSE	2	8	...
2351	1	park0004.jpg	FALSE	2	7	...

Figure 1. Crowdsourcing tasks are often delivered out of order when multiple workers accept them. Here, w_1 is asked to label a house image and is then asked to label a park image, instead of labeling images from the same group. Such context changes decrease performance and occur in many task domains that have a natural ordering.

CHI 2015, April 18–23 2015, Seoul, Republic of Korea

ACM 978-1-4503-3145-6/15/04 $15.00
http://dx.doi.org/10.1145/2702123.2702594

with 338 workers. Our results show both types of task interruptions can have significant harmful effects on completion time. Temporal interruptions can cause up to a 102% slowdown in task completion time (not counting the interruption itself), and contextual interruptions can slow down completion time by 57%. These findings suggest that both continuity and context in tasks is important for worker's productivity. We conclude with design recommendations that, by reducing interruptions, results in higher wages for workers and provide requesters with results more efficiently.

RELATED WORK

Research on workflows considers the costs and benefits of *interruptions*, or disruptions during the execution of a task. Interruptions decrease the performance of a worker shortly after an interruption [10]. This costs varies depending on the nature of the work being done, the type of interruption, and the worker's environment. Interruptions that are closely aligned with the worker's ongoing task are likely to be more disruptive regardless of how long they are, even if workers are prepared for them (which is common for microtasks) [8]. The costs of interruptions are based on cognitive load [1, 6], and workers perform poorly on the interrupting task as well [4]. We quantify the these effects in the context of microtasks.

We define interruptions not only as disruptions during execution of a single task, but also when workers are disrupted between microtasks.When workers are disrupted, they may forget some critical information and have to repeat a part of their work. For example, people who dial a phone while driving have to take moments to reexamine the road and adjust their course during interruptions [9]. In the case of microtasking, one might imagine the repeated sense-making that workers may have to do as they interleave and interrupt different varieties of tasks. They may have to re-learn how to complete a task or re-develop the necessary contextual expertise. On the other hand, tasks that underutilize cognitive resources can cause boredom that also reduces performance [13]. Continuous workflow models [11] hold the possibility of balancing worker's arousal and fatigue.

Since crowd workers are generally unknown to the requester and join or leave workflows frequently, typical task designs ensure that every task can be completed by a different, untrained and unexperienced worker. Accordingly, most work in crowdsourcing has focused on decomposable problems such as writing, editing [2], and image description [3], among others. Existing workflows focus on obtaining quality results by introducing redundancy and verification steps (e.g., answer agreement or the find-fix-verify pattern [2]).

While this approach maximizes the flexibility of the workforce by not requiring prior experience or long working periods, it disregards benefits like worker experience and memory. Because subsequent tasks are unrelated, the contextual disruptions cause a loss in specialization. Prior work has shown that despite often completing dozens of tasks per hour, workers remember task-specific details [12]. This means that discrete tasks often fail to leverage the experience workers gain over the course of completing multiple tasks [7]. Additionally, discretizing microtasks can cause a temporal disruption between the submission of one task and loading a subsequent task, which might cause workers to lose interest, move to another task completely, or earn less money for their time.

MOTIVATION

Prior research suggests that interruptions may be detrimental to workers, but workers may choose such workflows anyway. For workers, one of the draws of microtask work is flexibility. Workers might *prefer* to switch between different tasks. This means that while workflow literature may suggest this would lead to decreased performance and a likelihood of decreased pay, the variety of tasks may provide value to workers. However, workers on Mechanical Turk are known to have a selection bias towards tasks with more assignments [5].

Survey of Mechanical Turk Workers

To better understand the worker-centric factors that affect microtasking, we conducted a survey of 100 Mechanical Turk workers. The survey asked workers about their motivations, frequency of work, task or workflow preferences, and break-taking habits. 55 respondents identified as male. 55 were aged 18-29, 34 aged 30-39, and 11 aged 40+. 85 had a university degree. 58 workers identified Mechanical Turk as a major source of income.

Worker Habits

Workers spent an average of at least four hours per day ($M = 4.47$, $SD = 2.76$) working on Mechanical Turk tasks. There was a significant difference between the working hours of those doing tasks for pocket money, as part of their living, and as a job: 3.5 (41 respondants), 5 (34 respondants), and 5.41 (21 respondants) hours on average, respectively ($F(2, 97) = 4.99$, $p < 0.01$). 64 respondents reported that they took breaks, although we found no relationship between why workers work and break-taking. Those who took breaks said they worked for 1.24 hours on average ($SD = 2.68$) and then took a break averaging 16.6 minutes ($SD = 16.9$). Thus, many workers work for long enough periods to complete many different tasks.

Of the workers who did not take breaks, 64% mentioned concerns that breaks would decrease their earnings. 11% cited issues with "flow" when transitioning between tasks, 14% cited being satisfied with natural breaks between HITs, and 11% cited time or task demands preventing them from breaking.

Our study also suggests that workers remember the kinds of HITs they complete even long after they did them, suggesting the potential for task specialization. Our results showed that 78% of participants were able to describe the HIT they did right before taking the survey, and 74% could describe a HIT they really enjoyed form the past. Many workers described HITs they completed days, weeks, or months ago. This agrees with prior work showing that workers are able to retain and apply task information to improve their performance in future tasks [12]. Workers cited novelty, ease, speed, and repeatability as traits common in preferred tasks.

EMPIRICAL EFFECTS ON WORKERS

In the previous section, we presented evidence that workers are interested in sequential microtasking and actively seek

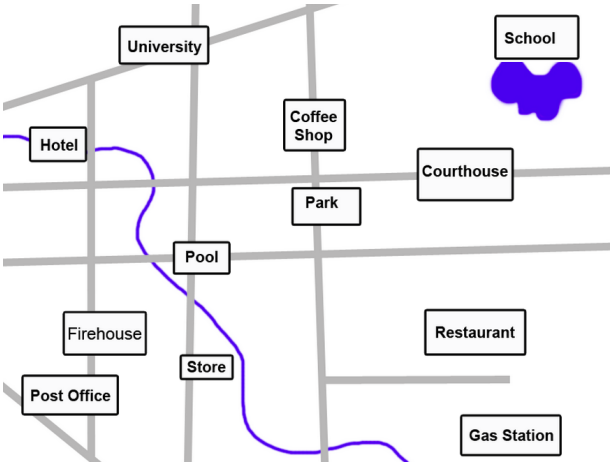


Figure 2. One of the map configurations generated for our trials. Workers were only able to see roughly $1/5^{th}$ of the map at a time through their viewport, meaning they had to scroll to find their target. Since each worker saw the same map for each of the primary search tasks, it was possible to learn the locations of the buildings over time.

it. With this in mind, we explored two common types of interruptions they may experience on a microtask platform: *temporal*, where a delay is added between tasks of the same type, and *contextual*, where different tasks are interleaved with tasks of the same type.

Experimental Setup

We created a task that asked workers to identify places on a map. Each map was larger than the user's viewport, forcing them to scroll to find a target (Figure 2). We generated a different map per worker that remained consistent across tasks so that they could remember places and geography. While we randomly placed the landmarks, all maps had the same landmarks distributed evenly in random locations. We used textual labels instead of graphical icons for clarity and to avoid cultural bias.

We varied the amount of time or type of context-based interruption. Since our tasks adhere well to the classic microtask model in which no prior expertise or context is required, workers had to eventually be able to get the correct answer to continue. Because of this, our measures focused on how long it took workers to achieve the task, rather than the degree to which they were correct.

We collected responses from 338 unique Mechanical Turk workers, and measured how long it takes them to complete tasks in different conditions and report how throughput is affected in terms of work lost. Workers were recruited half during the day in the U.S., and half during the day in India, and no recruitment filters were used. Task latency is our key measure because it captures how prepared workers were upon seeing a new task, and how their recall of the task improved their performance.

Control Task

In the control task, workers were first prompted to find and click on a particular landmark. Subsequent tasks utilized the same map but asked workers to find different landmarks. For example, a worker might be asked to click on the park. After correctly clicking on the park, they would be asked to click on the school. We expect that as workers are asked to use their map more and more, they will become more familiar with the layout of the map, and thus be able to find targets more quickly.

Temporal Interruptions

Traditional microtask interfaces often force workers to pause between tasks as a new task loads. To understand the effect of this delay, we modified the amount of time a worker had to wait between successfully completing one task and being given their next task so that it is not instantaneous as it was in the control task (C, $N = 57$). We used two delay lengths: 10 seconds (C_{short}, $N = 71$) and 30 seconds (C_{long}, $N = 67$). To avoid measuring workers time away from the task, we alert workers when the task is ready, and only measure from the time they actively begins the task. We did not find a significant difference between the average time workers spent finding landmarks in C and in C_{short} ($t(70) = 1.19$, $p = 0.24$)[1]. However, longer, 30-second breaks do have a significant effect ($t(66) = 3.40$, $p < 0.05$, Figure 3).

Contextual Interruption

On microtask platforms like Mechanical Turk task designers often provide contextually unrelated tasks to workers performing a HIT of a particular type. For example, workers captioning short audio segments from a larger recording are not necessarily given sequential pieces, but instead might be given a disjoint next piece available when they are ready for it (we discuss how this may be corrected later). To measure the effects of these contextual interruptions, we asked workers to complete a different task between each successive map task.

In the first condition (C_{map}, $N = 84$), workers who successfully identified a landmark were prompted to identify a new landmark situated on a *different* map. In the second condition (C_{image}, $N = 59$), workers who successfully identified a landmark were prompted to complete the unrelated task of image labeling (analogous to interleaving a different task in a workflow). After providing a short description of an image, workers were prompted to find another landmark on the same map as before.

For the image description distractor task, we did not find a significant difference between C and C_{image} ($t(58) = 0.31$, $p = 0.76$). As we expect from prior work, this is likely because the image description task was short and relatively disjoint from the map identification task. It does not appear to interfere with participants' performance in locating landmarks. However, the condition where workers were interrupted with a new map (C_{map}) showed a significant effect versus C ($t(83) = 4.39$, $p < 0.01$). Since this interruption was similar to the main task, it was more likely to interfere with workers' knowledge. Interestingly, this suggests that interleaving tasks is actually less of a potential detriment to workers' workflow than task design decisions. This fits with the broader picture of worker behavior discussed earlier.

[1]Since our data is not normally distributed, but is strongly log-normal, we apply a log-correction step below before running all significance tests.

DISCUSSION

Our results demonstrate that there can be a notable difference in performance when microtasks are presented in different workflow contexts. While the individual effects we measured initially look modest, their cumulative effect in a microtask setting can be quite large. For example, comparing the performance of participants who were shown the same kind of map landmark identification task, but with different maps interleaved as disruptions (condition C_{map}), it took 2.02 times as long on average to complete each task because participants' working knowledge and context were potentially disrupted. In our delay condition (C_{long}), even if the 30 second delay itself is completely factored out, it has the potential side effect of slowing the work conducted during active periods by a factor of 1.57, meaning workers would only get 63.7% of the work done compared to the baseline.

Our trials do not provide a comprehensive analysis of the types of tasks on crowdsourcing platforms today. However, they do provide insight into the potential effects of context and sequence delay in a microtask setting. Additionally, our experiments directly evaluation a common type of task: finding visual information (e.g., finding entries on a receipt). We believe this informs future designs for crowdsourcing tasks.

Resulting Design Guidelines

Condition C_{long} suggests that delays in task flows can seriously impact workers' performance well beyond the time lost simply because of the delay. Workers not only get bored and direct their attention elsewhere, but they may also forget their working context and need to rebuild their understanding of the task. We suggest *prefetching future tasks* in order to minimize potential delays between tasks. One simple way to do this is to simply load the next task in a hidden iframe. This has the advantage of working across a wide variety of different backend architectures. Although workers will take breaks when they feel necessary, this makes sure that the next task will load nearly instantly when they are ready.

Condition C_{map} suggests that contextual switches between tasks can confuse and interfere with workers' performance.

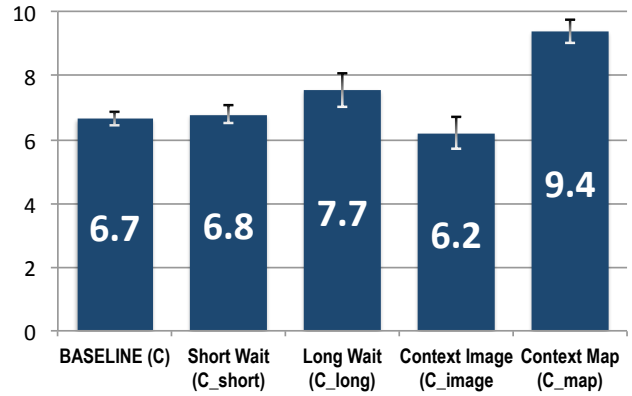


Figure 3. Task completion time increases when a worker completes different interleaved instances of a similar problem or waits for the longer period of time.

If workers are working within a batch of tasks, we suggest presenting them in a logical sequence. While there are many technical ways to achieve this, one simple approach that we use that works reasonably is to use a secondary sort by id or timestamp offset by an integer tied to the particular worker, i.e. sorting by $(id + MD5(workerid)) \% n$, where n is the total number of data records. This approach has the effect of starting each worker at a different random offset in the data.

CONCLUSION

In this paper, we experimentally demonstrated the impact of temporal and contextual interruptions on microtask workers. Our findings from 338 workers indicate that these interruptions can hurt worker performance. We concluded with a set of specific guidelines to help reduce these costs in practice.

ACKNOWLEDGMENTS

This work was supported by National Science Foundation #IIS-1149709, an Alfred P. Sloan Foundation Fellowship, and two Microsoft Research Ph.D. Fellowships.

REFERENCES

1. Adamczyk, P. D., and Bailey, B. P. If not now, when?: the effects of interruption at different moments within task execution. In *CHI 2004*.
2. Bernstein, M. S., et al.Soylent: a word processor with a crowd inside. In *Proc. UIST*, pages 313–322, 2010.
3. Bigham., J. P., et. al. Vizwiz: nearly real-time answers to visual questions. In *Proc. UIST*, pages 333–342, 2010.
4. Cabon, P., Coblentz, A., and Mollard, R. Interruption of a monotonous activity with complex tasks: effects of individual differences. In *Human Factors Society*, 1990.
5. Chilton, L. B., Horton, J. J., Miller, R. C., and Azenkot, S. Task search in a human computation market. In *HCOMP 2010*.
6. Cutrell, E., Czerwinski, M., and Horvitz, E. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. In *Proc. INTERACT*, pages 263–269, 2001.
7. Dow, S., Kulkarni, A., Klemmer, S., and Hartmann, B.. Shepherding the crowd yields better work. In *Proc. CSCW*, pages 1013–1022, 2012.
8. Gillie, T. and Broadbent, D.. What makes interruptions disruptive? a study of length, similarity, and complexity. *Psychological Research*, 50:243–50, 1989.
9. Jansset, C. P., and Brumby, D. P. Strategic adaptation to performance objectives in a dualtask setting. *Cognitive Science*, 34(8):1548–1560, 2010.
10. Kreifeldt, J. G., and McCarthy, M. E. Interruption as a test of the user-computer interface. In *Proc. MC*, 1981.
11. Lasecki, W. S., Murray, K., White, S., Miller, R. C., and Bigham, J. P. Real-time crowd control of existing interfaces. In *Proc. UIST*, pages 23–32, 2011.
12. Lasecki, W. S., White, S., Murray, K. I., and Bigham, J. P. Crowd memory: Learning in the collective. In *Proc. Collective Intelligence*, 2012.
13. Pattyn, N., Neyt, X., Henderickx, D., and Soetens, E. Psychophysiological investigation of vigilance decrement: boredom or cognitive fatigue? *Physiology and Behavior*, 93(1–2):369–378, 2008.