Association for Information Systems
AIS Electronic Library (AISeL)

ACIS 2013 Proceedings

Australasian (ACIS)

2013

Crowdsourcing, Cognitive Load, and User Interface Design

Bahareh Rahmanian
The University of Sydney, brah8754@uni.sydney.edu.au

Joseph G. Davis
The University of Sydney, joseph.davis@sydney.edu.au

ACIS 2013
RMIT MELBOURNE

Information Systems: Transforming the Future

24th Australasian Conference on Information Systems, 4-6 December 2013, Melbourne

Proudly sponsored by

ACIS 2013 Principal Sponsor

Crowdsourcing, Cognitive Load, and User Interface Design

Bahareh Rahmanian
Joseph G. Davis
School of Information Technologies
The University of Sydney
Sydney, Australia
Email: brah8754@uni.sydney.edu.au
joseph.davis@sydney.edu.au

Abstract

Harnessing human computation through crowdsourcing offers a new approach to solving complex problems, especially those that are relatively easy for humans but difficult for computers. Micro-tasking platforms such as Amazon Mechanical Turk have attracted large, on-demand workforces of millions of workers as well as hundreds of thousands of job requesters. Achieving high quality results and minimizing the total task execution times are the two of the main goals of these crowdsourcing systems. Drawing on cognitive load theory and usability design principles, we study the effects of different user interface designs on performance and the latency of crowdsourcing systems. Our results indicate that complex and poorly designed user interfaces contributed to lower worker performance and increased task latency.

Keywords

Crowdsourcing, Cognitive Load, User Interface

INTRODUCTION

Crowdsourcing has been discussed under various labels, including open innovation, collective intelligence, human computation, mass collaboration, distributed problem solving, and user-generated content, among others. It involves the harnessing of the collective knowledge and intelligence of a large number of individuals to generate solutions to relatively complex problems. Crowdsourcing has been reported to be efficacious in a variety of problem solving activities and turns out to be more effective than traditional computational approaches for some classes of problems. Human inputs are acquired and aggregated over the internet for solving problems that are relatively easy for people but difficult for computers, especially in areas such as image analysis, speech recognition, and natural language processing.

The Amazon Mechanical Turk (AMT or MTurk) and Crowdflower are examples of platforms that implement microtask-based crowdsourcing. These enable problem requesters to contract and interact with an on-demand, global workforce through a web-based user interface. Monetary reward is the main incentive and workers try to earn as much money as they can in short periods of time. Job requesters have to accept and pay workers for results or reject workers' results without paying them. In the case of tasks which solicit people's opinions, it is not possible to check all responses from workers and reject low performance results. Requesters tend to be willing to pay more for high quality crowd inputs. From both research and practical perspectives, it is important to identify the critical factors that affect crowdsourcing system's performance and to create interfaces that can promote better improved worker performance. .

Another important factor for requesters in crowdsourcing tasks is time. To make crowdsourcing tasks closer to real time, there need to be mechanisms to help workers find the tasks easily, complete them, and return the results as quickly as possible. If workers don't feel motivated to do the task because of the amount of reward or the task design, task completion time or crowdsourcing system latency will be increased.

Much of previous research on MTurk has tended to focus on factors that affect the motivation and creativity of workers and on cheating detection methods. There have not been many studies that deal with the impact of visual design of the tasks' interface on workers' performance and crowdsourcing system's latency. The usability of the software and user interface that are part of the MTurk platform can potentially affect worker satisfaction levels and costs incurred by the requesters. While many researchers have studied usability of systems in software design (Juristo et al. 2007; Liu and Ma 2010; Seuken 2010), and the effects of cognitive load and its integration with human computer interaction (HCI) concepts on user interface design (Antle and Wise 2013; Huang et al. 2009), there are a very few studies that have addressed the effects of user interface design on crowdsourcing using the MTurk platform.

It is our contention that the design of the interfaces through which the workers perform the human computation and related tasks has a significant effect on the performance and the time taken to complete the tasks. Drawing on cognitive load theory and usability design principles, we report on the design and preliminary results of two experiments that tested the effects of different user interface designs on performance and system latency in the context of crowdsourcing.

BACKGROUND AND RELATED WORK

The phrase 'crowdsourcing' was originally coined by Howe who defined it as "… the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call" (Howe 2006). It has evolved over the years into a range of endeavours including open innovation, distributed human computation, prediction markets, crowdfunding, and crowdservicing, to name a few (Davis 2011). We focus on the latter in which complex tasks are broken down into a large number of microtasks which are assigned to a large number of online workers through platforms such as Mechanical Turk.

Several researchers have attempted to categorize crowdsourcing systems based on multiple characteristics and behaviours of the tasks and people. (Wightman 2010) classified crowdsourcing systems based on the level of competitiveness of the tasks and motivation drivers into four categories. Erickson et al. applied five themes on four major categories of crowdsourcing (productivity, innovation, knowledge capture marketing/ branding) and suggested a framework (Erickson 2012; Erickson et al. 2012) which has some overlap with Yuen's four categories (voting systems, information sharing, game and creative systems) (Yuen et al. 2011).

In Amazon Mechanical Turk (MTurk), requesters can post their tasks. Workers sign into the system, search for their desired tasks, accept and solve the tasks and send the results back to MTurk. Micro-tasks on MTurk are referred to as HITs (Human Intelligence Tasks) and are grouped into HITGroups. Each requester can assign the same HITs to more than one worker. Some HITs are just a single task while others can be a collection of micro-tasks, such as providing labels for images or annotating text.

MTurk provides web-based interfaces for both requesters and workers to implement microtask-based crowdsourcing. Workers can login, search for tasks and perform the job. Requesters can chose between the web-based user interfaces (UI) to create simple HITs and collect results, or create more complex HITs using specialized UIs using Amazon's API for MTurk. MTurk APIs are provided in a variety of programming languages.

Regardless of the approach used to create HITs (web UI or API), all tasks are shown in an iframe inside workers' main web interface page. In order to view the HIT and complete the task, workers need to scroll within this iframe. This limited HIT design environment highlights the importance of a good UI design which is likely to affect the quality of results provided by workers and/or latency of crowdsourcing system.

Crowdsourcing Worker Performance

Since workers' performance influences overall crowdsourcing task performance, several researchers have studied aspects of worker performance. Monetary incentive is the primary motivation in MTurk. A few studies have investigated the effect of increasing the reward to get higher quality results. (Franklin et al. 2011) found that increasing monetary rewards decreased the time required for the HIT to be picked by workers. Another study by (Faridani 2011) implied that demand for task increased as a result of increased reward. However, in some crowdsourcing applications, higher rewards did not necessarily translate into higher quality responses (Buhrmester et al. 2011; Franklin et al. 2011; Mason and Watts 2009) . Cases have been reported in which increased payment resulted in reduced demand for the tasks, since high reward usually signifies higher complexity tasks (Franklin et al. 2011). Based on these studies, increasing reward by itself may not necessarily lead to improved results. It is important to assign a proper reward for the HITs in order to achieve lower response time and higher quality results. As well, research by (Franklin et al. 2011; Kittur et al. 2008) have shown that adding precise instructions to HITs and providing cheat detection mechanisms within the HIT directly have positive effects on worker performance. These mechanisms can also help to decrease gamers who try to cheat.

While the amount of monetary rewards and HIT instruction design can potentially affect workers' performance, there is not enough research on the impact of HIT UI design on workers' performance. As mentioned in the foregoing, the main mechanism of communication between workers and MTurk is the HIT's web-based UI. (Khanna et al. 2010) studied the impact of usability and UI design of MTurk on low-income workers in India. They discovered that varying skills are needed to do most of MTurk tasks and highlighted the need for designing better mechanisms to match tasks with workers capabilities. They also pointed out that complexities of user interface design and task instructions prevented their target group of workers from completing the task.

In the context of this research, we will investigate its impact on workers' performance by drawing on cognitive load theory and its extensions.

Latency in crowdsourcing systems

One major challenge in crowdsourcing systems is latency. The time interval between sending jobs to crowdsourcing platforms (e.g. MTurk) and receiving responses from workers can be broken down into two components. The first component, T1, is the time between sending the HITs to MTurk crowdsourcing platform, until some workers find your HITs, feel motivated to solve them, and start solving the problem. When workers start completing the HIT, it takes T2 time (the second component) for them to complete the job and send the results. The sum of T1 and T2 time is referred to as the total execution time (TET) of the crowdsourcing job. The whole process of sending HITs and receiving responses from workers may take minutes to days depending on the HITs design and the specified rewards. To make crowdsourcing applications near real-time, (Bigham et al. 2010) designed a mechanism called quikTurkit which recruits workers and keeps them busy with other available HITs until the required HIT arrives. The workers accept the actual HIT as it arrives and send back the responses. quikTurkit also uses search engine optimization techniques. As an example, quikTurkit posts more HITs than what is actually required and sends the alternate HITs by different titles or rewards. Using these mechanisms, quikTurkit tries to keep the posted HITs on the first page of the search results. By applying quikTurkit, they were able to receive their responses almost in real-time and at a low cost (Bigham et al. 2010).

In related research which sought to reduce latency of crowdsourcing systems, (Bernstein et al. 2012) proposed two techniques to get responses in just two seconds. First they defined a retainer model in which crowds are paid to wait until the actual task arrives. Unlike quikTurkit which keeps workers busy, users are free to do other HITs while waiting. When the actual task arrives, they are alerted and notified by sound. Rapid refinement is their second technique which seeks for early agreement on multiple responses to decrease overall amount of time needed to produce the desired result.

Even though previous research has attempted to design mechanisms to speed up recruitment and HIT selection process of crowdsourcing tasks, the impact of HIT UI design on TET of crowdsourcing tasks has not received adequate research attention.

THEORETICAL BACKGROUND: COGNITIVE LOAD

Cognitive load refers to the amount of mental resources required to process a given task; the higher amount of information needed to process a task, the more cognitive load the task has. Humans' mental resources are limited and when the amount of information and instruction for a task exceeds this limit, learning will be inhibited and performance will decrease. Sweller (Sweller 1988) described a model of cognitive load and distinguished three distinct memory types: sensory memory, working memory, and long-term memory.

Recent studies have focused on cognitive load and suggested that the limited working memory is the critical bottleneck in human information processing. Through these studies three type of cognitive load is distinguished by Cognitive Load Theory (CLT), intrinsic, extraneous and germane. Intrinsic cognitive is related to the level of expertise of a learner and defines by the intrinsic complexity of information that is to be learned (Bannert 2002; Sweller et al. 1998).Extraneous cognitive load is defined by any cognitive load associated with the way the task can be carried out and caused by activities that are irrelevant to the task (Ayres and Sweller 2005). (Paas and Merrienboer 1994) had found that variation of worked example types support construction of schema, but at the same time increase cognitive load. This type of cognitive load introduced as germane cognitive load.

CLT provides a basis to predict user performance when using different user interface designs, it also gives guidelines to minimize cognitive load in the design of user interfaces. In typical educational systems, monitoring and lowering cognitive load leads to increased student learning ratios. CLT research has also addressed techniques for decreasing extraneous cognitive load for these systems (Reis et al. 2012) and tried to design new interfaces that effectively minimizes student's cognitive load. Applying these findings to educational system's UI will help students focus on the learning task and learn efficiently. It also has been studied that using principles of user-centred design and CLT, leads to minimized extraneous cognitive load of the task (e.g. User input planning, minimize interruptions by eliminating unnecessary features, applying split-attention effect, redundancy effect, and modality effect learning technique) (Erry et al. 2006; Feinberg and Murphy 2000; Oviatt 2006).

Educational systems are not the only systems in which UI design directly affects user performance. While in crowdsourcing systems humans (workers) play the main role in solving problems, their performance directly affects the overall crowdsourcing tasks' quality. One of the disadvantages of MTurk as a crowdsourcing platform is its limitation in HIT interface environment. As previously mentioned, MTurk provides two separate web-based interfaces for requesters and workers. The third interface, which is the subject of this study, is HITs interface. This is the interface workers deal with to do the crowdsourcing task, and is to be shown in an iframe inside the MTurk's web page. This limited visual space of the HITs makes designing HIT UI an important consideration which can directly affect crowdsourcing task performance.

It is very common in crowdsourcing tasks to ask more than a worker to perform the same task. The final solution is then created by aggregating responses. If the quality of responses produced by each worker is low, requesters have to send more tasks and collect more results to come to a proper solution. In this situation crowdsourcing task will cost more money for requester. By increasing workers' performance and avoiding low quality results, the overall cost of crowdsourcing task will be decreased.

RESEARCH QUESTIONS AND HYPOTHESIS

As noted before, increasing performance and decreasing latency are two main goals in all crowdsourcing tasks. Requesters want their crowdsourcing tasks to be completed in minimum time with maximum quality results. HITs' UI design is an important aspect of crowdsourcing tasks. We address the following research questions:

- Does Cognitive Load Theory design principles help in designing improved interfaces for crowdsourcing tasks?
- Does the design of UIs impact workforce performance and productivity?

One of the CLT design suggestions is eliminating unnecessary distracting features in UI. If there are too many unnecessary features in a UI, more of the working memory will be wasted dealing with these features. It has been studied that if unnecessary features are eliminated, user's cognitive load will be minimized and result in higher learning ratio in educational software (Oviatt 2006). For the context of this research we will examine the same design principle and its effect you performance of results produced by workers in our crowdsourcing task with H1 hypothesis.

Another part of our study is to investigate the effect of HIT UI design on total execution time of the crowdsourcing task. We will try to answer if a poor HIT UI will increase crowdsourcing system latency or not.

The specific hypotheses are stated below:

H1: Lowering extraneous cognitive load by eliminating unnecessary features from HIT UI design will result in higher quality responses from workers.

H2: In the same task with similar reward, the complexity of HIT's UI has a negative effect on the total execution time (TET).

RESEARCH METHODOLOGY AND DESIGN

We describe two experiments that were performed to test the hypotheses. In Experiment1 we test H1 and study the impact of different UI designs on workers performance which directly affects cost of crowdsourcing task. This task we chose is an image ranking task. It involves ranking ten images based on their similarity to a given query image. The task involves visual information processing for which the quality of the user interface is particularly critical. For this experiment, we designed three different UIs based on ranking, direct sorting (drag and drop) and rating.

First UI design is called Rank UI design. In this type of design, we asked workers to compare ten images with the query image and rank them based on their similarity to the query image. Workers are asked to assign a number between 1 and 10 to each image indicating the position of the image in ranked list.

Second UI design is called Sort UI. In this type of UI, users have to click and move each image to create and visually create ranked list of images. In this UI design moving each image causes the whole list to be moved.

In the last UI design, called Rate UI, we asked workers to give a score between 1 and 5 based on the similarity of each image to the given query image. If the image is very similar to the query image they can assign score 5, and if it is not similar they can assign score 1 to it.

To test the second hypothesis we designed our second experiment (Experiment2). In this, the crowdsourcing task is to define a category for number of images. We designed two UI designs for HITs, Type1 and Type2. In this experiment we study the effect of HIT UI design on TET of crowdsourcing task and system latency.

Datasets

Experiment 1 involves assessing the performance of the workers in the ranking tasks for which a gold-standard is needed. We decided to use Corel-Princeton Image Similarity Benchmark Dataset for this reason. In this dataset, for each query image, similar images and their (gold standard) similarity score are provided. For our experiment we selected 6 query images and randomly select 10 similar images based on each query image. Aggregated ranking provided by workers were compared against the gold-standard ranking.

The task for the second experiment deals with image categorization for which we used a categorized image dataset. Caltech-256 data set was selected for this experiment. This dataset consists of more than 30,000 images categorized into 256 folders. Each folder has a category name.

EXPERIMENT 1: IMAGE RANKING

For Experiment1 we created and posted several HITs using the three UIs (Rank, Sort and Rate) that we designed. HIT structure for this experiment is:

- $0.05 reward for all three types of HITs
- instructions for users to do the task
- added time stamps to the design of the HIT to detect workers that just clicked and did not do the task carefully
- added a box to collect user comments

Figure 1 describes the system that was designed to create the HITs and collect and store the results obtained from the workers. MTurk makes it possible for workers to view the HIT in preview mode before accepting it. But in our experiment we just show a simple preview description and not the full HIT. When workers accept a HIT, the corresponding page is created on remote host and is shown to workers. Screenshots for each UI design are not included due to space constraints.

Users send their results back to MTurk using “Submit” button we provided on each page and then we collect results using our program and make them ready for analysis. (Figure 1)

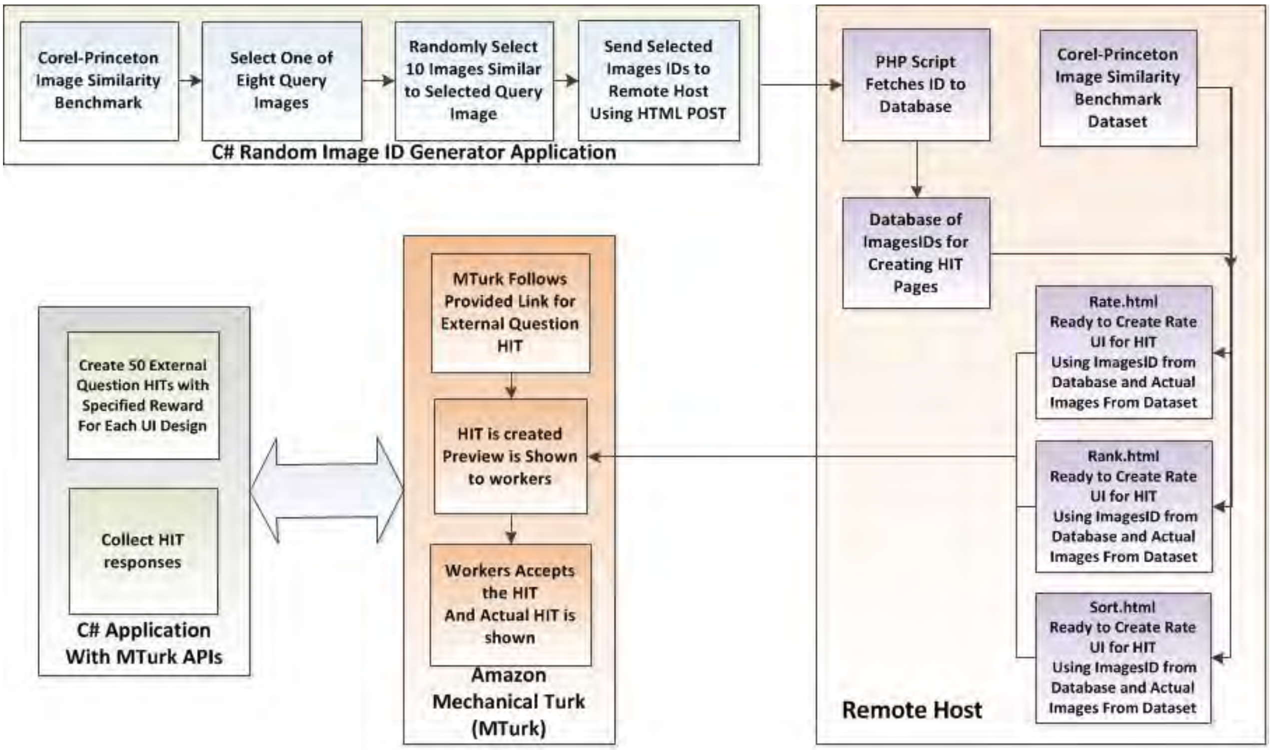

Figure 1: Experiment1

Rank UI design

For the Rank interface, we provide workers 10 images and asked them to assign a number between 1 and 10 to each image according to their similarity to the given query image. Value 10 means the given image is the most similar image to the query and value 1 means the image is the least similar image to the query image. Workers have to select a number for each image and they cannot use each value more than once. In this task design users had to compare 10 images with the query image and rank each image not only based on its similarity to the query image but also based on the degree of similarity to other images of the query image.

Sort UI design

In the this interface design for ranking images, we used some JQuery UI functions to create a draggable list of images and asked workers to sort images by their similarity to the given query image using drag-n-drop functionality of the HTML page. (Screenshots are available)

Rate UI design

In the Rate user interface design, once again we provided 10 randomly selected images from Corel-Princeton dataset. In this task we asked workers to rate the similarity of each image to the given query image. They were asked to assign a number between 1 and 5 according to similarity of each image to the given query image, 5 for high similarity and 1 for low similarity. In this task workers have to provide a rate for all images and they are allowed to use each number more than once.

Unlike the Rank method on which workers had to compare all images to provide rank for them, in this task they were able to focus on each image and rate its similarity to the query image.

Analysis and Results

We ran this experiment for 6 image categories from the Corel-Princeton Image Similarity Benchmark (airplane, car, flower, fruit, horse and model) and for each category we created 50 HITs for each of the three methods (Rank, Sort and Rate). In total, 350 HITs created for each UI design for $17.5. Total number of HITs created for this experiment is 1050HITs and $52.5 cost.

Since workers' responses for Rank and Sort methods are ranked lists, we aggregated them using Scaled Footrule Aggregation (Dwork et al. 2001). For the Rate method, we aggregated rates for each image by computing weighted average on rates given by workers, and then sorted the list according to this new calculated rate and created ranked list. To see how aggregated ranked lists provided by these three methods are close to gold standard rank. We calculated the distance between aggregated results and the gold standard ranking using Spearman's ρ correlation metric.

Results show that average difficulty levels of UI designs reported by workers are very close (Table1).

Table1: Difficulty of three UI designs reported by workers

	Rank	Sort	Rate
Average Difficulty Level	2.69	2.60	2.60

The ranking UI design with higher rank correlation coefficient with gold standard ranking has better performance. Results show that Rho rank correlation coefficient of the results produced using Rate UI design is higher than the other two methods. This implies that ranked list produced by users from Rate user interface is more similar to gold-standard ranked list created by professionals and relatively using Rank UI design leads to higher performance results (Table 2).

Table2: Distance between gold-standard ranking and experiment1 results

		Airplane Dataset	Cars Dataset	Flower Dataset	Fruit Dataset	Horse Dataset	Model Dataset
Rank UI	Spearman's ρ rank correlation	0.54	0.51	0.66	0.58	0.8	0.12
Sort UI	Spearman's ρ rank correlation	0.79	0.84	0.86	0.59	0.77	0.23
Rate UI	Spearman's ρ rank correlation	0.80	0.88	0.91	0.73	0.91	0.32

EXPERIMENT 2: IMAGE CATEGORIZATION

Second experiment's goal is to study the impact of UI design on TET with MTurk. We designed two different UIs for defining categories of images. Similar to Experiment1, HITs are sent to MTurk with $0.05 rewards. Caltech256 dataset is used for selecting categories and corresponding images. The system we used to create HITs and collect and store results is described in Figure 2.

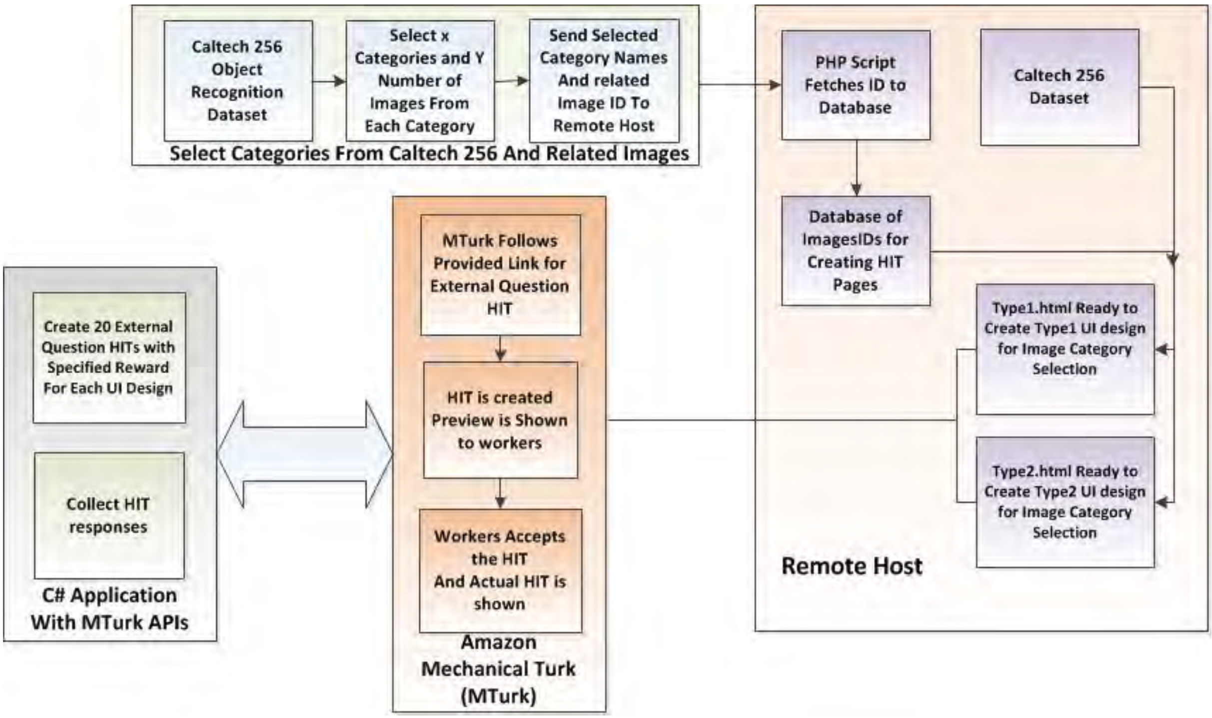

Figure 2: Experiment2

Type1 UI design

We put radio buttons for categories under each image and asked workers to select one category for each image.

Type2 UI design

Creating more complex user interface is goal of Type 2 UI. For this purpose, we first put all images at the top of the HIT webpage and in the bottom of the page we asked users to select images IDs which belong to a specific category. Due to limited space in MTurk's main HIT page, users have to scroll up and down to select image IDs for each category.

Analysis and Results

We ran this experiment twice for each UI design. On the first run, 12 images are selected from 5 different categories. We created 20 HITs for each UI. For Type1 UI design, it took 7 hours to have 20 completed HITs and just 3 workers returned the HIT, but for Type 2 UI design it took more than 16 hours to have 20 completed HITs and 12 workers rejected the task (Table3).

Table 3. Experiment 2 First Run Results; 12 Images From 5 Categories

	Type1 UI	Type2 UI
Total HITS	20	20
Total Cost	$1	$1
Average Task Time (s)	76	139
TET	7Hours, 20 Mins	16Hours, 40Mins

Number of rejected HITs	3	12

Users provided nearly 100% correct answers in both UI types. However the average completion time for Type1 is less than Type2. In Type 2 more workers rejected HITs, meaning that workers were not interested to do the Type2. This increased TET of the task and higher TET results in increased latency of crowdsourcing task.

In second run, we increased the number of images and categories. 21 images were selected from 8 different categories and 20 HITs created for each UI. This time it took 24 hours to have 20 completed HITs for Type1, but for Type2 after 36hours we received just 11 completed HITs. Hence we terminated the task. Checking the number of workers who did not complete the HIT shows that in Type1, 7 workers accepted the task but did not completed it and returned the HIT; and for Type2 62 workers rejected the task (Table 4).

Table 4. Experiment 2 Second Run Results; 21 Images From 8 categories

	Type1 UI	Type2 UI
Total Completed HITS	20	11
Total Cost	$1	$1
Average Task Time (s)	137	244
TET	23Hours, 50 Mins	38Hours
Number of Rejected HITs	7	62

These results show the importance of designing a task UI which makes workers more interested in. If workers are not interested in HIT UI design, they will reject the HIT. As a result, the latency of crowdsourcing will increase.

DISCUSSION

Results of our Experiment 1 pointed out the importance of lowering extraneous cognitive load of UI design and its effect on performance of results produced by workers. While all the three parts of experiment cost the same, using the interface which results in higher performance responses from workers will make the crowdsourcing task more affordable.

Taking a closer look at these three UI designs we can say that in Rank UI design users have to compare the whole 10 images with each other and the query image to find a rank for each image. While number of images to compare is more than Miller's magic number 7+_2 (Miller 1956), we believe it imposes higher cognitive load on the task resulting in lower performance and poorer results.

In the Sort UI design, users have to move images to come to ranked list and moving one single image makes the whole list move. These movements of the images in the page distract the user from the original task and put more cognitive load on the task. Also number of user clicks is higher in this UI design which is not suggested by CLT. Unnecessary distracting features and high number of clicks put more cognitive load on the task and tend to poor results from users.

We believe the reason that workers perform better with the Rate UI design is they can focus on each image by itself and assign a similarity score. This reduces the number of comparisons from 10 to 2 resulting in lower cognitive load. These results suggest that if the task has higher intrinsic cognitive load, poorly designed UI design with high extraneous cognitive load can have a negative effect on workers' performance.

In Experiment 2 we studied the impact of UI design on TET of a crowdsourcing task. In this experiment increased cognitive load and higher complexity of the UI design did not affect workers' performance but it contributed to reduced willingness levels to accept and finish the task. If workers do not like to accept and finish the task, requesters will not receive their desired number of responses. This means higher system latency and also contributes to increased probability of incomplete crowdsourcing tasks.

Results of our experiments highlight the demand for more research on UI design of MTurk HITs with aspect of cognitive load and usability. We examined impact of UI design on two visual crowdsourcing tasks. Cognitive load aspect of HIT design in textual tasks can also be studied. In our future work we will use the findings of this study to design crowdsourcing tasks.

CONCLUSION

In light of the limitations of the generic user interface in MTurk, it is important to design HIT UIs that reduce poor quality results and increase worker productivity. This has the potential to reduce the execution time of the crowdsourcing task. In this paper we tried to study the impacts of user interface design of HITs in the MTurk crowdsourcing platform on workers' performance and total execution time. Our experiments show that designing a HIT UI with the goal of reducing the cognitive load will help workers focus on the task and perform better. In some crowdsourcing tasks it is not possible to differentiate false results and reject workers' responses (like our image ranking task), so requesters have to pay all workers. We showed that in such tasks it is possible to have higher quality results by eliminating the factors that lead to workers poor performance, with the same cost. This means we have more cost effective crowdsourcing task.

We also investigated the effect of UI design on the demand for task. Our results show that MTurk workers prefer to accept tasks with less complex UI designs. If the user interface is perceived to be complex from the workers' point of view, they are less likely to accept the crowdsourcing tasks. As a result, it takes more time to complete the task and the crowdsourcing system's latency will increase. To have a crowdsourcing system closer to real time, we suggest spending more time on HIT UI to design an UI with less complexity.

Results of our experiments show that by spending more time on HIT UI design, requesters can achieve high quality results in shorter time. The results can help develop guidelines for making crowdsourcing tasks more efficient with less latency.

REFERENCES

Antle, A., and Wise, A. 2013. "Getting down to details: Using theories of cognition and learning to inform tangible user interface design" *Interacting with Computers* (25:1).

Ayres, P., and Sweller, J. 2005. "The Split-Attention Principle in Multimedia Learning," in *The Cambridge handbook of multimedia learning*, New York, NY, USA: Cambridge University Press, pp. 135–146.

Bannert, M. 2002. "Managing cognitive load—recent trends in cognitive load theory," *Learning and Instruction* (12:1), pp. 139–146.

Bernstein, M., Karger, D. R., Miller, R. C., and Brandt, J. 2012. "Analytic Methods For Optimizing Realtime Crowdsourcing," in *Preceeding of Collective Intelligence conference -CI 2012*, .

Bigham, J. P., White, S., Yeh, T., Jayant, C., Ji, H., Little, G., Miller, A., Miller, Robert, Miller, Robin, Tatarowicz, A., and White, B. 2010. "VizWiz," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology - UIST '10*, New York, New York, USA, October 3, p. 333.

Buhrmester, M., Kwang, T., and Gosling, S. D. 2011. "Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data?," *Perspectives on Psychological Science* (6:1), pp. 3–5.

Davis, J. 2011. "From Crowdsourcing to Crowdservicing," *IEEE Internet Computing* (15:3), pp. 92–94.

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. 2001. "Rank aggregation methods for the Web," in *Proceedings of the tenth international conference on World Wide Web - WWW '01*, New York, NY, USA, pp. 613–622.

Erickson, L. B. 2012. "Leveraging the crowd as a source of innovation," in *Proceedings of the 50th annual conference on Computers and People Research - SIGMIS-CPR '12*, New York, NY, USA, May 31, p. 91.

Erickson, L. B., Petrick, I., and Trauth, E. M. 2012. "Organizational uses of the crowd," in *Proceedings of the 50th annual conference on Computers and People Research - SIGMIS-CPR '12*, New York, NY, USA, May 31, p. 155.

Erry, C., Ginns, P., and Pitts, C. 2006. "Cognitive load theory and user interface design: Making software easy to learn and use" , Retrieved 8 March 2013 From http://www.ptgglobal.com/PDFArticles/Cognitive%20load%20theory%20and%20user%20interface%20design%20Part%201%20v1.0.pdf

Faridani, S. 2011. “What ’ s the Right Price? Pricing Tasks for Finishing on Time,” *Human Computation: Papers from the 2011 AAAI Workshop (WS-11-11)* , pp. 26–31.

Feinberg, S., and Murphy, M. 2000. “Applying Cognitive Load Theory to the Design of Web-Based Instruction,” in *Proceedings of 2000 Joint IEEE International and 18th Annual Conference on Computer Documentation (IPCC/SIGDOC 2000)*, , pp. 353–360.

Franklin, M. J., Kraska, T., Xin, R., Ramesh, S., and Kossmann, D. 2011. “CrowdDB: Answering Queries with Crowdsourcing,” in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data SIGMOD ’11*, New York, NY, USA, pp. 61–72.

Howe, J. 2006, June. “Crowdsourcing: A Definition,” Retrieved 4 October 2012 From http://www.crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html .

Huang, W. C. (Darren), Trotman, A., and Geva, S. 2009. “A Virtual Evaluation Forum for Cross Language Link Discovery,” in *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, , pp. 19–20.

Juristo, N., Moreno, A. M., and Sanchez-Segura, M.-I. 2007. “Analysing the impact of usability on software design,” *Journal of Systems and Software* (80:9), pp. 1506–1516.

Khanna, S., Ratan, A., Davis, J., and Thies, W. 2010. “Evaluating and improving the usability of Mechanical Turk for low-income workers in India,” *Proceedings of the First ACM Symposium on Computing for Development - ACM DEV ’10* New York, New York, USA: ACM Press, p. 1.

Kittur, A., Chi, E. H., and Suh, B. 2008. “Crowdsourcing user studies with Mechanical Turk,” in *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI ’08*, New York, NY, USA, April 6, p. 453.

Liu, H., and Ma, F. 2010. “Research on Visual Elements of Web UI Design,” in *IEEE 11th International Conference on Computer-Aided Industrial Design & Conceptual Design (CAIDCD)*, Yiwu, China, pp. 428–430.

Mason, W., and Watts, D. J. 2009. “Financial Incentives and the ‘ Performance of Crowds ’,” in *Proceedings of the ACM SIGKDD Workshop on Human Computation-HCOMP ’09*, New York, NY, USA, pp. 77–85.

Miller, G. a. 1956. “The magical number seven, plus or minus two: some limits on our capacity for processing information. 1956.,” *Psychological review* (101:2), pp. 343–52.

Oviatt, S. 2006. “Human-Centered Design Meets Cognitive Load Theory: Designing Interfaces that Help People Think,” in *Proceedings of the 14th annual ACM international conference on Multimedia - MULTIMEDIA ’06*, New York, NY, USA, pp. 871–880.

Paas, F. G. W. C., and Merrienboer, J. J. G. Van. 1994. “Variability of Worked Examples and Transfer of Geometrical Problem-Solving Skills : A Cognitive-Load Approach c o,” *Journal of Educational Psychology* (86:1), pp. 122–133.

Reis, H. M., Borges, S. S., Durelli, V. H. S., Fernando de S. Moro, L., Brandao, A. a. F., Barbosa, E. F., Brandao, L. O., Isotani, S., Jaques, P. a., and Bittencourt, I. I. 2012. “Towards Reducing Cognitive Load and Enhancing Usability through a Reduced Graphical User Interface for a Dynamic Geometry System: An Experimental Study,” in *2012 IEEE International Symposium on Multimedia*, , December, pp. 445–450.

Seuken, S. 2010. “Hidden Markets : UI Design for a P2P Backup Application,” in *CHI2010 : Market Models for Q&A Services*, Atlanta, Georgia, USA, pp. 315–324.

Sweller, J. 1988. "Cognitive Load During Problem Solving: Effects on Learning," *Cognitive Science* (12)J. Sweller, pp. 257–259.

Sweller, J., Merrienboer, J. J. G. Van, and Paas, F. G. W. C. 1998. "Cognitive Architecture and Instructional Design," *Educational Psychology Review* (10:3), pp. 251–296.

Wightman, D. 2010. "Crowdsourcing Human-Based Computation," in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries - NordiCHI '10*, New York, NY, USA, pp. 551–560.

Yuen, M.-C., King, I., and Leung, K.-S. 2011. "A Survey of Crowdsourcing Systems," in *2011 IEEE Third Intl Conference on Privacy Security Risk and Trust and 2011 IEEE Third Intl Conference on Social Computing*, (Vol. 58) , pp. 766–773.

COPYRIGHT