CrowdScape: Interactively Visualizing User Behavior and Output

Jeffrey M. Rzeszotarski, Aniket Kittur
Human-Computer Interaction Institute
Carnegie Mellon University
{jeffrz, nkittur}@cs.cmu.edu

ABSTRACT
Crowdsourcing has become a powerful paradigm for accomplishing work quickly and at scale, but involves significant challenges in quality control. Researchers have developed algorithmic quality control approaches based on either worker outputs (such as gold standards or worker agreement) or worker behavior (such as task fingerprinting), but each approach has serious limitations, especially for complex or creative work. Human evaluation addresses these limitations but does not scale well with increasing numbers of workers. We present CrowdScape, a system that supports the human evaluation of complex crowd work through interactive visualization and mixed initiative machine learning. The system combines information about worker behavior with worker outputs, helping users to better understand and harness the crowd. We describe the system and discuss its utility through grounded case studies. We explore other contexts where CrowdScape's visualizations might be useful, such as in user studies.

ACM Classification:
H5.m. Information interfaces and presentations (e.g., HCI).

General terms:
Design, Human Factors, Measurement

Keywords:
Crowdsourcing, Visualization, Interfaces, Event Logging, User Behavior, Quality Control, Performance

INTRODUCTION
Crowdsourcing markets help organizers distribute work in a massively parallel fashion, enabling researchers to generate large datasets of translated text, quickly label geographic data, or even design new products [4,11,20]. However, distributed work comes with significant challenges for quality control. Approaches include algorithmically using tools such as gold standard questions that verify if a worker is accurate on a prescribed baseline, majority voting where more common answers are weighted, or behavioral traces where certain behavioral patterns are linked with outcome measures [4,6,10,16]. Crowd organization algorithms such as *Partition-Map-Reduce, Find-Fix-Verify,* and *Price-Divide-Solve* distribute the burden of breaking up, integrating, and checking work to the crowd [2,14,15].

UIST'12, October 7–10, 2012, Cambridge, Massachusetts, USA.

These algorithmic approaches can be effective in deterministic or constrained tasks such as image transcription or tagging, but they become less effective as tasks are made more complex or creative [7,13,14]. For example, subjective tasks may have no single "right" answer, and in generative tasks such as writing or drawing no two answers may be identical. Conversely, looking at the way workers behave when engaged in a task (e.g., how they scroll, change focus, move their mouse) rather than their output can overcome some of these challenges, but may not be sufficiently accurate on its own to determine which work to accept or reject [16]. Furthermore, two workers may complete in a task in very different ways yet both provide valid output.

We present CrowdScape, a system that supports the evaluation of complex and creative crowdwork by combining information about worker behavior with worker outputs through mixed initiative machine learning (ML), visualization, and interaction. By connecting multiple forms of data, CrowdScape allows users to develop insights about their crowd's performance and identify hard workers or valuable products. The system's machine learning and dynamic querying features support a sensemaking loop wherein the user develops hypotheses about their crowd, tests them, and refines their selections based on ML and visual feedback. CrowdScape's contributions include:

- An interface for interactive exploration of crowdworker results that supports the development of insights on worker performance by combining information on worker behavior and outputs;
- Novel visualizations for crowdworker behavior
- Novel techniques for exploring crowdworker products
- Tools for grouping and classifying workers
- Mixed initiative machine learning that bootstraps user intuitions about a crowd.

In the remainder of this paper we will describe the technical details of the visualization system and illustrate its utility through several grounded case studies.

QUALITY CONTROL IN CROWDSOURCING
Low quality work is common in crowdsourcing markets, comprising up to an estimated one third of all submissions

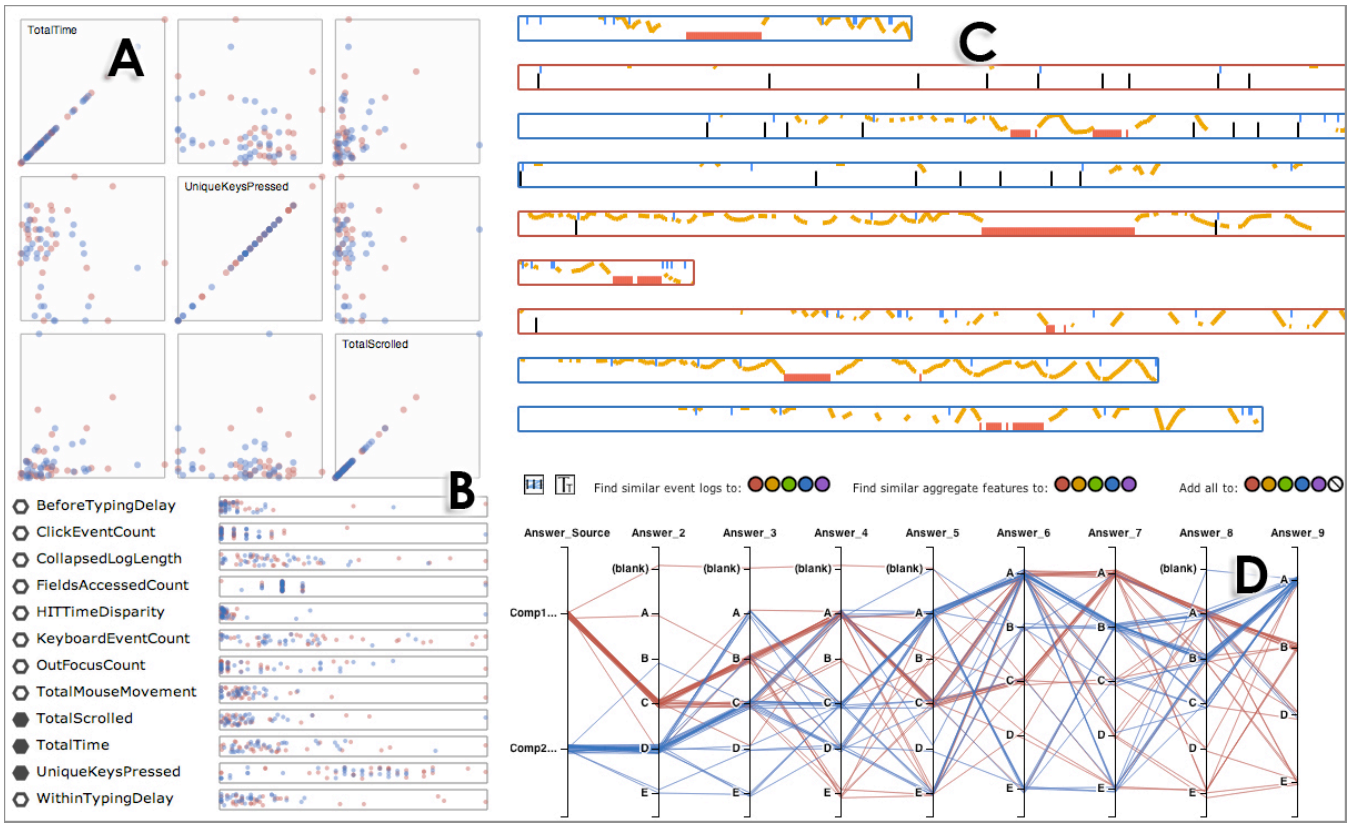


Figure 1: The CrowdScape interface. (A) is a scatter plot of aggregate behavioral features. Brush on the plot to filter behavioral traces. (B) shows the distribution of each aggregate feature. Brush on the distribution to filter traces based a range of values. (C) shows behavioral traces for each worker/output pair. Mouseover to explore a particular worker's products. (D) encodes the range of worker outputs. Brush on each axis to select a subset of entries. Next to (B) is a control panel where users can switch between parallel coordinates and a textual view of worker outputs (left buttons), put workers into groups (colored right buttons), or find points similar to their colored groups (two middle button sets).

[2]. As a result, researchers have investigated several ways of detecting and correcting for low quality work by either studying the post-hoc pool of outputs or the ongoing behavior of a worker.

Post-hoc output evaluations

Validated 'gold standard' questions can be seeded into a task with the presumption that workers who answer the gold standard questions incorrectly can be filtered out or given corrective feedback [4,8]. In the case of well defined tasks such as transcribing a business card, it is easy to insert validated questions. However, in more complex tasks such as writing validation questions often do not apply. Other researchers have suggested using trends or majority voting to identify good answers, or to have workers rate other workers' submissions [2,4,18]. While these techniques can be effective (especially so when the range of outputs is constrained) they also are subject to gaming or majority effects and may completely break down in situations where there are no answers in common such as in creative or generative work.

Another method researchers have employed relies on organizing and visualizing crowd workflows in order to guarantee or work towards better results. Turkomatic and CrowdWeaver use directed graph visualizations to show the organization of crowd tasks, allowing users to better understand their workflow and design for higher quality [13,15]. CrowdForge and Jabberwocky use programmatic paradigms to similarly allow for more optimal task designs [1,14]. These tools can provide powerful ways of organizing and managing complex workflows, but are not suited to all tasks and require iteration to perfect.

Behavioral traces

Another line of research suggests that looking at the manner in which workers complete a task might provide enough information to make inferences about their final products [16,19]. For example, a worker who quickly enters tags one after the other may be doing a poorer job than a worker who enters a tag, pauses to glance back at the image, and then enters another. While harnessing these implicit behavioral features can be effective, it requires that at least some of the feature vectors be labeled by examining and evaluating worker outputs manually. CrowdFlower's analytics tools address this issue, aligning post-hoc outcomes such as gold standard question responses with individual workers in visual form [5]. As a result, this tool can surface general worker patterns, such as failing certain gold

questions or spending too little time on a task. However, without access to detailed behavioral trace data, the level of feedback it can provide to task organizers is limited.

Integrated Quality Control

While each of these categories has advantages and disadvantages, in the case of creative or complex work neither are sufficient alone. There may not be enough data to train predictive models for behavioral traces, or it may be difficult to seed gold standard questions. Yet, in concert both post-hoc output analysis and behavioral traces provide valuable complementary insights. For example, imagine the case of image tagging. We may not have enough labeled points to build a predictive model for a worker who enters tags in rapid succession, but we may recognize that this worker submits two short tags. Another worker may also enter the same two tags and share a similar behavioral trace. From this we might posit that those two tags are indicators of workers who behave in a slipshod manner. By combining both behavioral observations and knowledge of worker output, we gain new insight into how the crowd performs.

CROWDSCAPE

CrowdScape, as illustrated in Figure 1, is built on top of Mechanical Turk, a crowdsourcing market, capturing data from both the MTurk API in order to obtain the products of work done on the market and Rzeszotarski and Kittur's Task Fingerprinting system in order to capture worker behavioral traces [16]. CrowdScape uses these two data sources to generate an interactive data visualization which is powered by Javascript, JQuery, and D3.js [3].

Scenario

Imagine a requester has two hundred workers write short synopses of her collection of YouTube physics tutorials so that she can pick the best ones to use as her video descriptions. She turns to CrowdScape to parse through her pool of submissions. Since she added code to log worker behavior and has stored her collection of worker outputs, she inputs them into the interface and begins exploring her data.

She wants to be sure that people actually watched her video before summarizing, so she locates the 'Total Time' aggregate feature (a behavioral trace of actual time spent working). She then brushes the scatter plot, selecting workers who spent a minimum reasonable amount of time on the task. The interface dynamically updates all other views, filtering out several non sequiturs and one-word summaries in the worker output panel.

She now looks through a few worker's logs and end products by hovering over their behavioral trace timelines for more details. She finds several that submitted good descriptions of her videos, so she places them into the same colored group. She uses the mixed-initiative machine learning feature to get suggestions for submissions similar to her labeled group of 'good' submissions. The list reorders, and she quickly adds some similarly good-sounding summaries to her final list. After repeating the process several times, she feels she has a good list of candidates, and exports her submissions so that she can add them to YouTube.

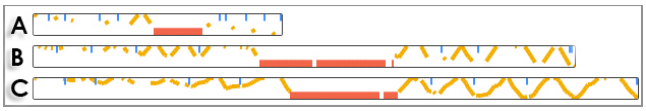


Figure 2: Workers clicking radio buttons while referring to a source passage at the top of their view.

WORKER BEHAVIOR

CrowdScape utilizes two data sources: worker behavior and output. Each has important design considerations for interaction and visualization. In the case of worker behavior, there are two levels of data aggregation: raw event logs and aggregate worker features.

Individual Traces

Raw event logs measure worker behavior on a highly granular, user interface interaction level, providing time-series data for user mouse mouse movements, clicks, scrolls, keypresses, and focus changes. A key challenge in CrowdScape is representing this time series data in a way that is accurate yet easy to interpret and detect differences and patterns in worker behavior.

To address this challenge we developed a method for generating an abstract visual timeline of a trace. Our designs focused on promoting rapid and accurate visual understandings of worker behavior. We represent the time a worker takes to do certain tasks horizontally, and place indicators based on the different activities a worker logs. Through iteration we determined that representing keypresses, visual scrolling, focus shifts, and clicking provided a meaningful level of information. Representing mouse movement greatly increased visual clutter and in practice did not appear to provide useful information for the user. Keypress events are logged as vertical red lines that form blocks during extended typing, and help to differentiate behaviors such as copy-pasting versus typing. Clicks are blue flags that rise above other events so they are easily noticed. Browser focus changes are shown with black bars to suggest the 'break' in user concentration. Scrolling is indicated with orange lines that move up and down to indicate page position and possible shifts in user cognitive focus. To make it easy to compare workers' completion times we used an absolute scale for the length of the timeline; this proved more useful than normalizing all timelines to the same length as it also allowed accurate comparison of intervals within timelines.

The colors and flow of the timelines aim to promote quick, gestalt understandings of a user's behavior. For instance, compare the three timelines in Figure 2. *A* is a *lazy* worker who picks radio buttons in rapid succession. *B* is an *eager* worker who refers to the source text by scrolling up to it in between clicking on radio buttons and typing answers. *B*'s scrolling manifested in the U-shaped orange lines as they jump from the button area to the source text as well their

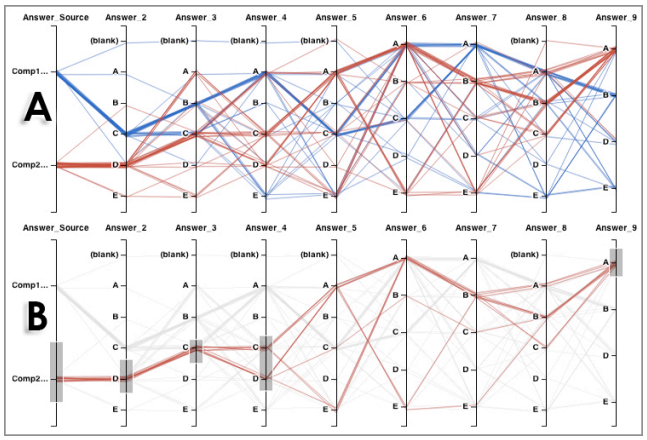


Figure 3: Two views of submission parallel coordinates for a text comprehension quiz. (A) shows all points while (B) uses brushing to show a subset.

keyboard entry. Such patterns manifest in other diligent workers within the same task (such as *C*).

To support larger scale exploration over hundreds or thousands of worker submissions we provide a means to algorithmically cluster traces. The user first provides a cluster of exemplar points such as the group of similarly behaving users in the earlier example (workers *B* and *C*). We compute the average Levenshtein distance from the exemplar cluster to each of the other workers' behavioral traces and order them based on their 'closeness'. This allows users to quickly specify an archetypical behavior or set of behaviors and locate more submissions that exhibit this archetype.

Aggregate Features

We also visualize aggregate features of worker behavioral traces. These have been shown to be effective in classifying the workers into low and high performing groups, or identifying cheaters. Making these numerous multi-dimensional features understandable is a key challenge for CrowdScape. We first reduced the number of dimensions by eliminating redundant or duplicate features in favor of features shown to be effective in classifying workers in previous research [16]. This resulted in twelve distinct aggregate features.

Given our list of twelve features, we use a combination of 1-D and 2-D matrix scatter plots to show the distribution of the features over the group of workers and enable dynamic exploration. For each feature we use a 1-D plot to show its individual characteristics (Figure 1B). Should the user find it compelling, they can add it into a 2-D matrix of plots that cross multiple features in order to expose interaction effects (Figure 1A).

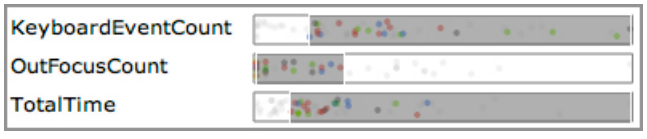


Figure 4: Brushing ranges of aggregate features

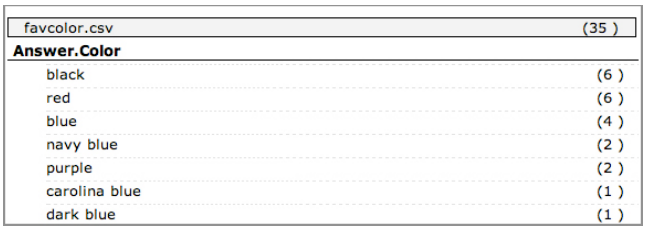


Figure 5: The text view of submissions for a survey. This view is useful if the parallel coordinates (Fig. 3) are saturated with singletons or large text entries.

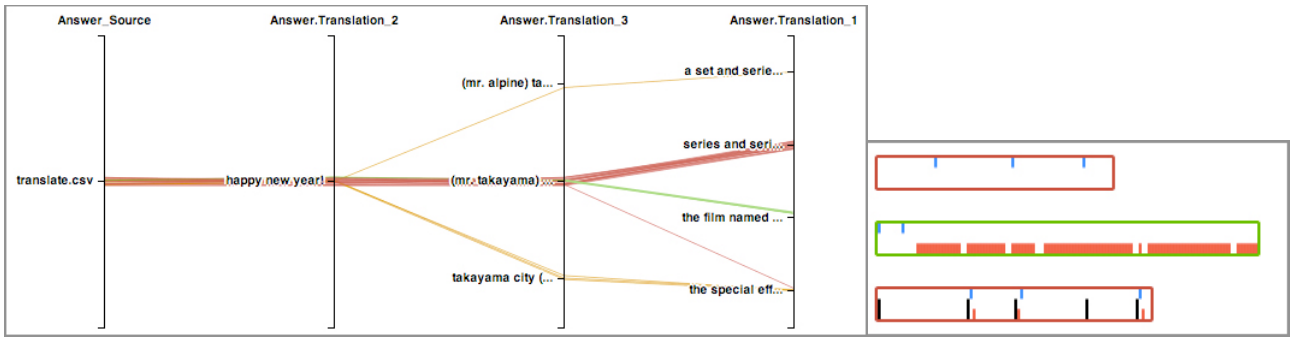
While these static visuals are effective at showing distributions and correlations, we further use dynamic querying to support interactive data analysis. Users can brush a region in any 1D or 2D scatter plot to select points, display their behavioral traces, and desaturate or filter unselected points in all other interface elements. This interactivity reveals multidimensional relationships between features in the worker pool and allows users to explore their own mental model of the task. For example, in Figure 4 the user has selected workers that spent a fair amount of time on task, haven't changed focus too much, and have typed more than a few characters. This example configuration would be useful for analyzing a task that demands concentration.

Yet, it still may be difficult to spot multi-dimensional trends and explore the features of hundreds or thousands of points. As a result we provide a means to cluster submissions based on aggregate event features. Similar to the ML behavioral trace algorithm, the user provides exemplars, and then similar examples are found based on distance from a centroid computed from the selected examples' aggregate features. The system computes the distance for all non-example points to the centroid and sorts them by this similarity distance. This allows users to find more workers whose behavior fits their model of the task by triangulating on broad trends such as spending time before typing or scrolling.

WORKER OUTPUT

Though visualizing worker behavior is useful, users still require an understanding of the final output a worker produced. One challenge for CrowdScape is representing worker output in a meaningful way. For a scale larger than ten or twenty workers, serially inspecting their contributions can be intractable and inefficient. Instead, we chose to focus on two different characteristics of worker submissions.

The first characteristic is that worker submissions often follow patterns. For example, if a user is extracting text from a document line-by-line, the workers that get everything right will tend to look like each other. In other words, workers that get line 1 correct are more likely to get line 2 correct and so forth. These sorts of aggregate trends over multiple answer fields are well suited for parallel coordinates visualizations [9]. For each answer section, the system finds all possible outcomes and marks them on parallel



Figures 6 and 7: Figure 6 shows the parallel coordinates for a 21 translations of 3 sentences. Note that only one translator (green) was successful. Red and orange translators copies from machine translation services. Observe the green translator's markedly different behavioral trace.

vertical axes. Each submission then is graphed as a line crossing the axes at its corresponding answers. Figure 6 shows one such trend, highlighting many workers who answer a certain way and only a few workers who deviate. Figure 3 shows a far more complex relationship. To help disambiguate such complex output situations, the system allows for dynamic brushing over each answer axis. This allows a user to sift through submissions, isolating patterns of worker output (Figure 3B).

Not all tasks generate worker output that is easy to aggregate. For writing a review of a movie, few if any workers will write the exact same text (and those that did would likely be suspect). The system provides a means to explore the raw text in a text view pane, which users can view interchangeably with the parallel coordinates pane. The text view pane shows answers sorted by the number of repeat submissions of the same text. For example if one were to ask workers to state their favorite color, one would expect to find lots of responses to standard rainbow colors, and singleton responses to more nuanced colors such as "fuschia" and "navy blue" (Figure 5). The text pane view is also linked with the other views; brushing and adding items to categories is reflected through filtering and color-coded subsets of text outputs, respectively.

INTEGRATING BEHAVIOR AND OUTPUT

While on their own behavioral traces and worker output visualizations can provide useful insights to crowd organizers, together they can provide far more nuanced information. For instance, imagine the case where users are translating a passage sentence-by-sentence. Worker agreement in this case may identify a cluster of identical good translations, but also a cluster of identical poor translations copy-pasted into translation software. Behavioral trace visualization can provide additional insights: the software group may show evidence of taking very little time on the task, or using copy-paste rather than typing. They may change focus in their behavioral traces. The typing group may show large typing blocks in their traces, delays of deliberation, and take longer to complete the task. Thus combining behavioral traces and worker outputs can provide more insight than either alone.

It is through dynamic querying and triangulation that CrowdScape helps users to develop mental models of behavior and output like described above. Dynamic queries update the interface in realtime as filters are applied and data is inspected [17]. Such interaction techniques augment user understanding through instantaneous feedback and enabling experimentation. Thus, by brush-selecting on the aggregate feature of time spent in CrowdScape, the parallel coordinate display of worker output as well as behavioral traces update accordingly. Picking one point highlights it in every axis at once. Even further, the interface supports assigning group identities to points using color. This allows users to color-code groups of points based on their own model of the task and then see how the colors cluster along various features.

This unity between behavior and output fosters insights into the actual process workers use to complete a task. Users develop a mental model of the task itself, understanding how certain worker behaviors correlate with certain end products. In turn, they can use this insight to formulate more effective tasks or deal with their pool of worker submission data.

CASE STUDIES

To illustrate the different use cases of CrowdScape, we posted four varieties of tasks on the Amazon Mechanical Turk crowdsourcing market and solicited submissions. We logged worker behavior (with explicit worker consent), recorded output, generated raw traces and aggregate features, and then imported them into CrowdScape for study.

Translation

CrowdScape reveals patterns in workers that help to unveil important answers that majority-based quality control may miss. We posted a task asking for workers to translate text from Japanese into English, figuring that lazy Turkers would be likely to use machine translation to more quickly complete the task. We chose three phrases: a conventional "Happy New Year" phrase which functioned as a gold standard test to see if people were translating at all, a sentence about Gojira that does not parse well in mechanical translators, and a sentence about a village that requires do-

Figure 8: Traces for two color survey workers

main knowledge of geography to translate properly. We had 21 workers complete this task at a pay rate of 42 cents.

After importing the results of the task into CrowdScape, one feature in the output of the workers is immediately revealed by the parallel coordinates interface of worker products in Figure 6. All workers passed our gold, translating 'Happy New Year" properly. However, 16 out of 21 workers submitted the same three sentences; this pattern is clearly delineated by the dark line of multiple submissions (red in the figure). Examining their submissions shows that they likely used Google Translate, which is able to translate the first two sentences properly, but stumbles on the Gojira film sentence. Another bold line at the bottom shows a grouping of workers who used a different machine translation service (orange).

Eliminating those two groups, two workers are left. The orange line at the top shows one such worker. Note that the grammatical errors in their third submission are rather similar to the red machine translation group, suggesting more machine translation. The alignment of the parallel coordinates helps to expose these patterns. We are left with only one worker who likely translated the task manually, producing a reasonably accurate translation of the final sentence. This is confirmed by their behavioral traces (the green bar in Figure 7), which show evidence of time spent thinking, lack of focus changes (e.g., to copy-paste to and from translation software), and significant time spent typing (as opposed to copy-pasting).

This case study demonstrates the power of CrowdScape in identifying outliers among the crowd. By examining the pattern of worker submissions, one can quickly hone in on unique behaviors or outputs that may be more valuable than common behaviors or submissions made by the crowd.

Picking a Favorite Color

CrowdScape can also support or refute intuitions about worker cognitive processes as they complete tasks. We posted a task asking workers to use an HSV color picker tool to pick their favorite color and then tell us its name. 35 workers completed the job for 3 cents each. With this task in mind, we developed the model that workers who spent a long time picking a color were likely trying to find a more specific shade than 'red' or 'blue' which are easy to obtain using the color picker. In turn, we posited that workers that identified a very specific shade were more likely to choose a descriptive color name since they went to the trouble.

As anticipated, CrowdScape showed that the three most common colors were black, red, and blue (Figure 5). In order to explore our theory about worker cognition, we

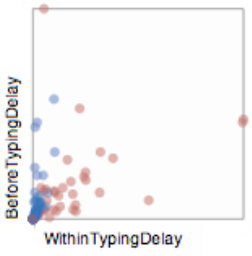


Figure 9: Scatter plot for workers who summarized and tagged (red) and only tagged (blue)

filtered submissions by the amount of time workers waited before typing in their color. This reduced the amount of submissions, revealing workers who wrote colors such as "Carolina blue", "hot pink", or "teal". The difference is evident in the workers' behavioral traces as well (Figure 8).

This case demonstrates that CrowdScape supports the investigation of theories about worker cognitive processes and how they relate to workers' end products. By simply following our intuition that more deliberation may suggest more descriptive colors, we were able to locate an interesting set of minority submissions.

Writing About a Favorite Place

CrowdScape supports feedback loops that are especially helpful when worker output is extremely sparse or variable. We asked 50 workers to describe their favorite place in 3-6 sentences for 14 cents each. No two workers provided the same response, making traditional gold standard and worker majority analysis techniques inapplicable. Instead, we explored the hypothesis that good workers would deliberate about their place and description and then write about it fluidly. This would manifest through a higher time before typing and little time spent between typing characters. After configuring the scatterplot matrix to pair the two aggregate features for typing delays (similar to Figure 9) we selected a region on the graph that described our hypothesis and were left with 10 selected points. By hovering over each one, we quickly scanned their responses, binning good ones into a group. We then used the machine learning similarity feature to find points that had similar aggregate behavioral features. We chose this over finding similar traces because workers in practice did not scroll, click, or change focus much. After we found points with similar features, we repeated the same process, quickly binning good descriptions. After one more repetition, we had a sample of 10 acceptable descriptions.

Our ending response set satisfices our goal of finding a diverse set of well-written favorite places. Descriptions ranged from the beaches of Goa, India, a church in Serbia, a park in New York, and mountains in Switzerland. By progressively winnowing our submissions by building a feedback loop using recommendations and binning, CrowdScape allowed us to quickly develop a successful final output set.

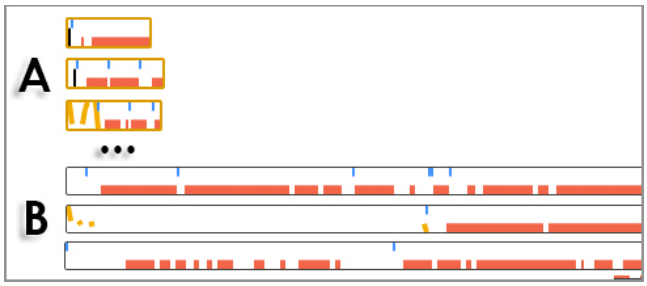


Figure 10: Traces for workers who only tagged videos (A) and for workers who tagged and summarized videos (B)

Tagging a Video

To explore this feedback loop in more detail, we had 96 workers tag science tutorial videos from YouTube for either 25 or 32 cents. Some workers also summarized the video, based on Kittur et al.'s design pattern for having easily monitored tasks that engage workers in task-relevant processing [12]. Binning the workers into two groups immediately shows that workers who only gave tags (in blue) spent less time than summarizers (in red) deliberating before and during their text entry (Figure 9).

The behavioral traces also expose another nuance in the pool of workers: some workers watch the whole video then type, other workers type while watching, and some seemingly don't watch at all. We first scrolled through the entire pool of traces, looking for telltale signs of people who skipped the video such as no focus changes (interactions with the flash video player) and little white space (pauses). After identifying several of these traces, we had the machine learning system generate similarity ratings for the rest of the traces based on the traces of our group of exemplars. This yielded several more similar cases where workers did not watch the video and instead added non-sequitur tags such as "extra", "super" and "awesome". Among these cases were some good submissions, suggesting that our initial insight that shorter traces might correlate to worse tags is incomplete. However, when examining strings highly *dissimilar* to our bad examples, they were almost universally good. This was extreme enough that we felt we could take the bottom half of the list of submissions sorted by similarity to the bad examples and have a sufficient set of good tags. Figure 10 illustrates the contrast between our bad exemplars and the set of good 'dissimilar' points.

This case demonstrates the sorts of detailed insights that CrowdScape promotes by unveiling the intersection between output and behavior. We were able to find evidence supporting our hypothesis that asking workers to summarize produced better tags, while also identifying an usable subset of valid outputs.

DISCUSSION, LIMITATIONS, AND FUTURE WORK

CrowdScape links behavioral information about workers with data about their output through an interactive visualization and mixed-initiative machine learning. While past work in crowdsourcing has focused on worker output or worker behavior alone, combining them has several advantages. In the case of purely generative tasks (such as describing one's favorite place), this combination helps a user quickly explore the space of worker submissions, facilitating their development a mental model of the behavior of workers who have good or bad outputs. This model helps them identify further good workers and output in a sort of positive feedback loop. Visualizing the process workers use to complete a task can contradict or reinforce our conception of the cognitive processes they use to complete a task, which in turn informs our understanding of the task's end products (as in the favorite color case). The video tagging case illustrates the value of mixing the two data sources together, permitting organizers to understand their crowd's collective action in greater detail.

These contributions may have use beyond crowdsourcing, for example in supporting studies of user interface design. By pairing the behavioral traces of users as they interact with an interface with outputs such as whether they completed the task successfully, CrowdScape could quickly unveil elements of tasks or designs that might be improved over iteration. This would enable deeper analysis of user study data, as well as dealing with some of the challenges inherent in remote user studies (in which users are typically unobserved).

There are some limitations to CrowdScape's current approach. In some situations it may not be possible to capture worker behavior easily; currently, CrowdScape is limited to online web pages in which Javascript can be inserted. Thus the approach may not be applicable to non-web interfaces, pages where the requester does not have access to inject Javascript, or when the worker blocks scripting (though the latter can be tested for). In the case of generative tasks like describing favorite places, the parallel coordinates view becomes saturated and the linear list can become quite long. Techniques for filtering the data (e.g., by behavior in the "favorite place" case study) can help alleviate this issue. Furthermore, it is unclear whether the set of aggregate features included will always be useful, and whether similarity based on distance will always provide useful feedback.

There are situations in which learning to use CrowdScape and employing it may not be appropriate. For instance, in cases where there are clear, consensus ground truths such as identifying a spelling error, CrowdScape does not provide a significant advantage compared to preexisting quality control measures. There may also be situations in which the behavioral traces analyzed are not very indicative of the way that work is done. For example, if users complete most of the task in a separate text editor and paste the text in at the end, their behavioral trace will contain very little info (other than focus changes and that someone pasted, which can itself be informative). Taken further, tasks that are completely offline or primarily cognitive (such as considering the next move in a game of chess) may not be amenable to the generalized approach espoused by

CrowdScape. Further work is needed to test the approach on a wider variety of tasks.

One other potential weak point of CrowdScape is in the detail level of the behavioral traces. Currently it relies on mouse movement, scrolling, keypresses, focus events, and clicks. This may not be sufficient for determining, for instance, where the fovea of the user is currently focused. However, we could design tasks to provide more detailed feedback in exchange for increasing obtrusiveness. For instance, we could only play a video if the mouse is hovered over it, allowing us to measure when it is playing. Or, we could require a text area to be clicked and held to show its actual text, allowing us to accurately estimate where and when the user is directing their attention. However, advantages in measuring user behavior more accurately must be balanced against increases in intrusiveness and decreases in efficiency.

CrowdScape aims to unify different quality control approaches, benefiting from their synergy. Gold standard and worker majority can help task organizers immediately evaluate worker output. Aggregate behavioral traces can help isolate target worker clusters. Individual traces can provide insight into actual worker behaviors. By unifying these approaches, CrowdScape allows users to develop and test their mental models of tasks and worker behaviors, and then ground those models in worker outputs and majority or gold standard verifications. Furthermore, CrowdScape's dynamic querying system permits the rapid analysis of large sets of data by providing immediate feedback to users. As crowd work becomes increasingly creative, collaborative, and complex, we hope that integrated systems such as CrowdScape will enable organizers to better understand and harness the nature of their crowd.

ACKNOWLEDGMENTS

We thank Paul André, Rebecca Gulotta, Lisa Yu, and our reviewers among many others for their feedback and support. This research was supported by NSF grants OCI-09-43148, IIS-0968484, IIS-1111124, and a grant from Carnegie Mellon's Center for the Future of Work.

REFERENCES

1. Ahmad, S., Battle, A., Malkani, Z., and Kamvar, S. The Jabberwocky programming environment for structured social computing. *In Proc. UIST '11*, (2011), 53-64.
2. Bernstein, M.S., Little, G., Miller, R.C., et al. Soylent: a word processor with a crowd inside. *In Proc. UIST '10,* ACM (2010), 313–322.
3. M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics 17,12*. (2011), 2301–2309.
4. Callison-Burch, C. Fast, cheap, and creative: evaluating translation quality using Amazon's Mechanical Turk. *In Proc. Empirical Methods in Natural Language Processing*. (2009), 286–295.
5. CrowdFlower. www.crowdflower.com. (2012).
6. Dekel, O. and Shamir, O. Vox populi: collecting high-quality labels from a crowd. *In Proc. 22nd Annual Conference on Learning Theory*, (2009).
7. Dow, S., Kulkarni, A., Klemmer, S., and Hartmann, B. Shepherding the crowd yields better work. *In Proc. CSCW '11*, (2011).
8. Downs, J.S., Holbrook, M.B., Sheng, S., and Cranor, L.F. Are your participants gaming the system?: Screening Mechanical Turk workers. *In Proc. CHI '10*, (2010).
9. Inselberg, A. and Dimsdale, B. Parallel coordinates: a tool for visualizing multi-dimensional geometry. *IEEE Visualization*, (1990), 361-378.
10. Ipeirotis, P.G., Provost, F., and Wang, J. Quality management on Amazon Mechanical Turk. *In Proc. ACM SIGKDD workshop on human computation*, (2010).
11. Ishikawa, S. Clickworkers interactive: towards a robust crowdsourcing tool for collecting scientific data. *Lunar and Planetary Institute*, (2012), 2-3.
12. Kittur, A., Chi, E., and Suh, B. Crowdsourcing user studies with Mechanical Turk. *In Proc. CHI '08*, (2008).
13. Kittur, A., Khamkar, S., André, P., and Kraut, R. CrowdWeaver: Visually managing complex crowd work. *In Proc. CSCW '12*, (2012).
14. Kittur, A., Smus, B., and Khamkar, S. Crowdforge: Crowdsourcing complex work. *In Proc. UIST '11*, (2011).
15. Kulkarni, A., Can, M., and Hartmann, B. Collaboratively crowdsourcing workflows with Turkomatic. *In Proc. CSCW '12*, (2012).
16. Rzeszotarski, J.M. and Kittur, A. Instrumenting the crowd: using implicit behavioral measures to predict task performance. *In Proc. UIST '11*, (2011).
17. Shneiderman, B. Dynamic queries for visual information seeking. *Software, IEEE*, (1994).
18. Snow, R., O'Connor, B., Jurafsky, D., and Ng, A.Y. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. *In Proc. Empirical Methods in Natural Language Processing*, (2008), 254–263.
19. Vanderaalst, W., Vandongen, B., Herbst, J., Maruster, L., Schimm, G., and Weijters, a. Workflow mining: a survey of issues and approaches. *Data & Knowledge Engineering 47*, 2 (2003), 237-267.
20. Yu, L. and Nickerson, J. Generating creative ideas through crowds: an experimental study of combination. *In Proc. ICIS*, (2011), 1-16.