Crowd-Powered Intelligent Systems

Walter S. Lasecki
ROC HCI Lab
Computer Science Department
University of Rochester
http://wslasecki.com
wlasecki@cs.rochester.edu

Abstract

Human computation provides a new resource for creating intelligent systems that are able to go beyond the capabilities of fully-automated solutions. Currently, machines struggle in many real-world settings because problems can be almost entirely unconstrained and can vary greatly between instances. Solving problems such as natural language understanding require artificial intelligence to first reach human levels, a goal that is still very far off. Human computation can help bridge the gap between what computers can do and users' ideal interactions, but is traditionally applied in an offline, batch-processing fashion. My work focuses on a new model of continuous, real-time crowdsourcing that enables *interactive* crowd-powered systems to be created.

Introduction

Intelligent systems make it easier for users to find and access information, complete tasks, and get feedback. However, current computational systems are limited by the abilities of machines to perceive and reason about the world. My work explores how to create systems that overcome these barriers by integrating human intelligence from dynamic groups (crowds) of people who can be recruited on-demand to quickly contribute small units of work. These people might be volunteers, full-time employees, or workers on micro-task marketplaces, such as Amazon's Mechanical Turk. Underlying these systems is a novel class of workflows and interfaces that allow people and machines to build on one another's abilities. My work has been the first to explore crowd-powered systems that are able to provide real-time responses *continuously* over multiple interactions.

In this paper, I will outline some of the systems that I have created during my Ph.D. work that exemplify key algorithms and design principles for systems that combine human and machine intelligence in ways that yield better performance than either one alone (Figure 1). My work has two main thrusts: creating access technologies for users with disabilities and general intelligence systems that go beyond what AI can do alone, and modeling crowdsourcing workflows that optimize for response speed and consistency.

Access Technology

Access technology provides users with cheaper, more readily available, and more easy-to-use tools for perceiving, understanding, and interacting with the world around them. However, classic (machine) computation often struggles most in exactly those domains that are most needed by users with disabilities, such as vision, natural language processing, and recognizing speech.

I have created systems that provide deaf and hard of hearing users with real-time captions (Lasecki et al. 2012), recognize real-word activities to help older adults and people with certain cognitive impairments live more independently (Lasecki et al. 2013a; 2014b), and make getting answers to visual questions much more efficient and usable for blind and low vision users (Lasecki et al. 2013b). The potential for creating systems that have a transformative effect on people's daily lives drives me to not only develop approaches that can satisfy access needs, but also to build and deploy them so that people can use these tools.

General Intelligent Systems

Using the same principles that make it possible to support access technologies using the crowd, we can also develop general-purpose intelligent systems that are able to interact with people more naturally than automated systems alone. For example, Chorus (Lasecki et al. 2013c) is a crowd-powered conversational assistant that is able to answer arbitrary user questions in natural language over multiple turns of conversation, and Glance (Lasecki et al. 2014a) allows researchers and other analysts to code video using natural language descriptions in minutes instead of days.

Models of Crowdsourcing

All of the work I will describe in this paper fits into a framework for interactive crowdsourcing called *crowd agents*. This model views a machine and workflow moderated collective as a single intelligent entity, but to achieve this, crowdsourcing workflows must be focused on ensuring quick, consistent, and reliable output. These ideas are developed throughout the systems in the following section.

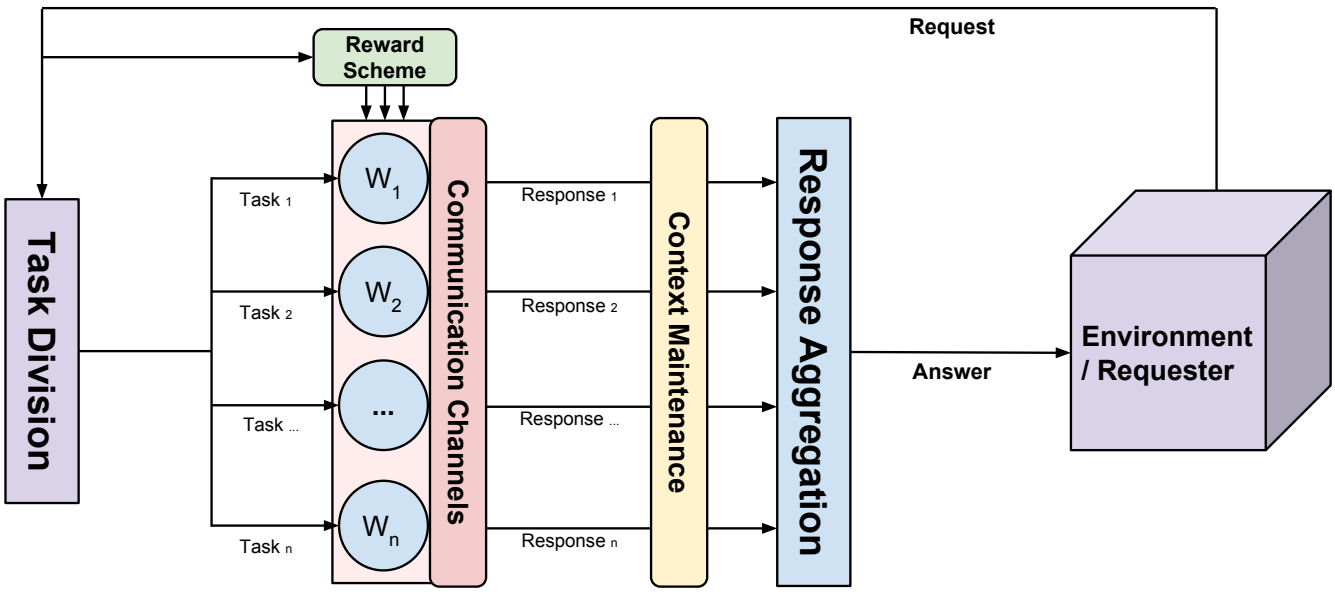

Figure 1: The architecture of a real-time crowd-powered system.

Systems

To demonstrate the power and generalizability of interactive crowd-powered systems, we have developed several systems that explore different key approaches to organizing collective intelligence processes in real-time.

Legion

Legion (Lasecki et al. 2011) is a system that enables groups of crowd workers to collectively control an existing user interface in real-time with no modification needed to the interface itself. This work was the first to introduce the continuous real-time crowdsourcing model. Prior work had investigated real-time crowdsourcing in the context of being able to quickly recruit members of the crowd to perform a single discrete task (Bernstein et al. 2010; Bigham et al. 2010). However, in many cases this limits the amount of feedback that workers can get regarding their inputs effect on the task itself. We expect that this leads to a decreased task learning rate for workers. Furthermore, for time-sensitive tasks that involve performing several small tasks, the overhead incurred by each worker having to accept a new task at every step might exceed the actual work provided.

Workers were asked to perform an interactive control task (such as driving a robot), so we need a model that would allow each worker to get feedback from the environment, while also leveraging the wisdom of the crowd. Legion does this by binning time into small windows (roughly 0.5-1 seconds), then implicitly using each workers input as a vote for a given action. The most effective strategy across the tasks we tested turned out to be weighting workers based on their current and past agreement with the rest of the crowd, then electing a leader for each time window. This avoided cases where vote splitting in the crowd caused the average decision to be incorrect. For example, when half the crowd wanting to drive a robot left to avoid an obstacle, and the other half wanting to drive to the right.

Legions use of input from different workers selected at different times allowed the final result to be a single, coherent control stream that could be forwarded to an existing interface. In effect, all of the workers collectively acted as a single worker, allowing the interface to be controlled without needing any special modification. Legion handles a diverse range of tasks, and has been tested in a variety of different interfaces and domains using tasks such as robot navigation through a physical maze, powering an intelligent assistive keyboard for motor impaired users, data-entry in a spreadsheet, OCR from an image of text, and games.

Scribe

Legion:Scribe (Lasecki et al. 2012; Lasecki, Miller, and Bigham 2013) provides real-time captions of speech using non-expert workers who each type part of what they hear. Currently, the two most common approaches to collecting real-time captions are professional stenographers who can cost as much as $200/hour, and automatic speech recognition, which does not provide usable captions in real settings.

Scribe aims to solve that by allowing multiple non-expert captionists to collaboratively caption speech in real-time (Figure 2). Professional stenographers train for years to be able to keep up with natural speaking rates that average about 150 words per minute. Non-expert workers are generally only able to caption about a third of this rate. However, Scribe is able to merge multiple non-expert captionists into a single reconstruction of the original speech by using an online multiple sequence alignment algorithm. This differs from the original model presented in Legion since instead of the final output being a selection of different worker inputs, Scribe actually merges the captions together to outperform what an constituent worker could do alone.

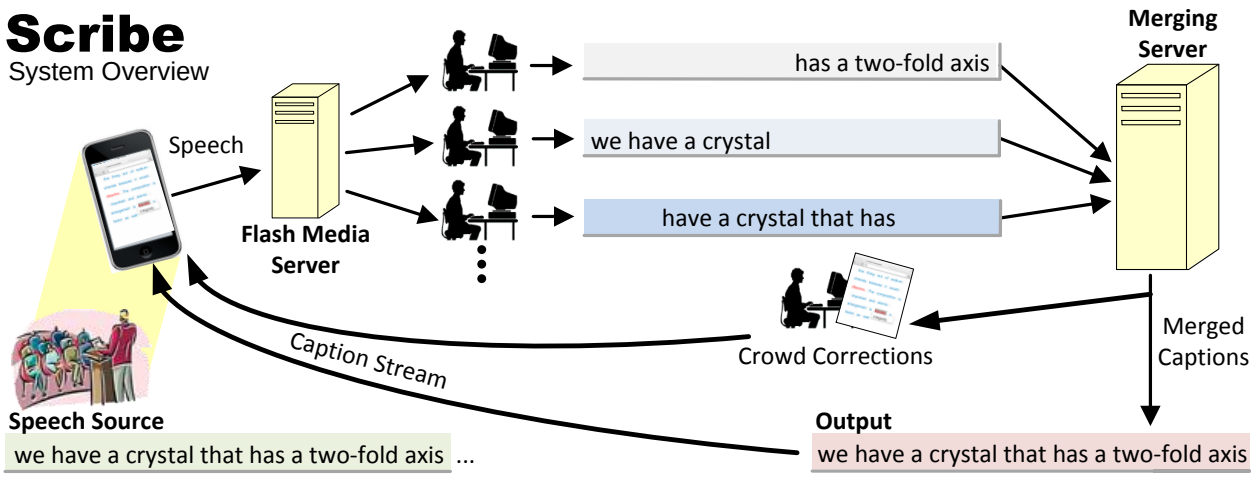

Figure 2: Scribe allows users to caption audio on their mobile device. The audio is sent to multiple amateur captionists who use Scribe's web-based interface to caption as much of the audio as they can in real-time. These partial captions are sent to our server to be merged into a final output stream, which is then forwarded back to the user's mobile device. Crowd workers are optionally recruited to edit the captions after they have been merged.

Glance

Behavioral researchers spend considerable amount of time coding video data to systematically extract meaning from subtle human actions and emotions. We introduced Glance to improve this process. Glance is a tool that allows researchers to rapidly query, sample, and analyze large video datasets for behavioral events that are hard to detect automatically. Glance takes advantage of the parallelism available in paid online crowds to interpret natural language queries and then aggregates responses in a summary view of the video data. Glance provides analysts with rapid responses when initially exploring a dataset, and reliable codings when refining an analysis (Figure 3).

Our experiments show that Glance can code nearly 50 minutes of video in 5 minutes by recruiting over 60 workers simultaneously, and can get initial feedback to analysts in under 10 seconds for most clips. We present and compare new methods for accurately aggregating the input of multiple workers marking the spans of events in video data, and for measuring the quality of their coding in real-time before a baseline is established by measuring the variance between workers. Glance's rapid responses to natural language queries, feedback regarding question ambiguity and anomalies in the data, and ability to build on prior context in follow-up queries allows users to have a conversation-like interaction with their data – opening up new possibilities for naturally exploring video data.

Chorus

Legion allowed an interface to be autonomously controlled by the crowd using natural language, but didn't allow the workers to respond to the user if the problem was underspecified, or if the crowd wanted to know which valid option to select. While many other previous crowdsourcing systems also provide end-users with the ability to make natural language requests to the crowd, most do not allow for multiple inputs in a turn, and none allowed for a dialogue with the crowd. Chorus was developed to allow users to hold a conversation with the crowd using an instant messenger style interface. To make this conversation understandable to users, Chorus encourages workers to respond as if the crowd were a single consistent conversational partner.

This consistency is a product of two main components. The first is a propose-and-filter step in which workers are asked to both propose responses to a workers input then see what others have proposed and vote for those answers they think are the most appropriate. Workers are encouraged to do this task correctly using a game theoretic mechanism that rewards workers for proposing or voting for an answer that is chosen by the crowd as correct, and effectively penalizing incorrect input.

The second component that Chorus uses to maintain consistent conversations is a working memory. This is similar to work we had originally performed using Legion, in which organizational learning was observed via implicit demonstration and observation during an hour-long navigation task. In Chorus, this idea of a collective memory of past events is made explicit in the form of a highlights window which shows what facts workers think are the most important to the current task. The ability of the crowd to remember as a group, by passing on knowledge to future workers points to potential applications of this model for other organizations. For example, customer service centers which want to provide a more consistent experience to customers (i.e. not forcing them to repeat their problem to every representative they talk to), even when individual employees are not always available for the entire session. Another important setting for this work is in the healthcare domain, where doctors must hand off patient charts during shift changes.

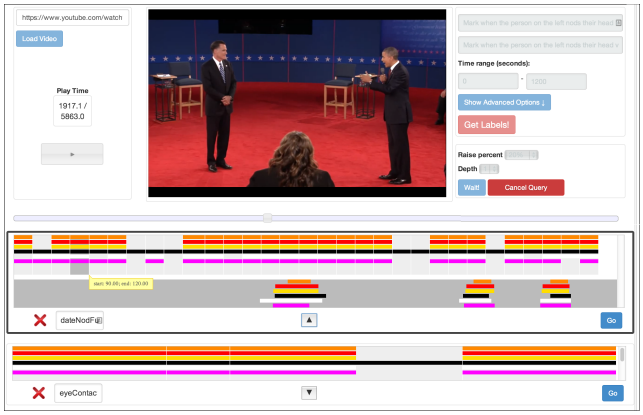

Figure 3: The Glance analyst user interface. Analysts can load a video from YouTube, ask if or when an event occurs in the video in natural language, and set parameters to control cost and speed. Crowd workers then process this query and return results in a fraction of the playtime of the video. These results are aggregated to simplify the answer for the analyst, and make it more reliable than one worker's answer.

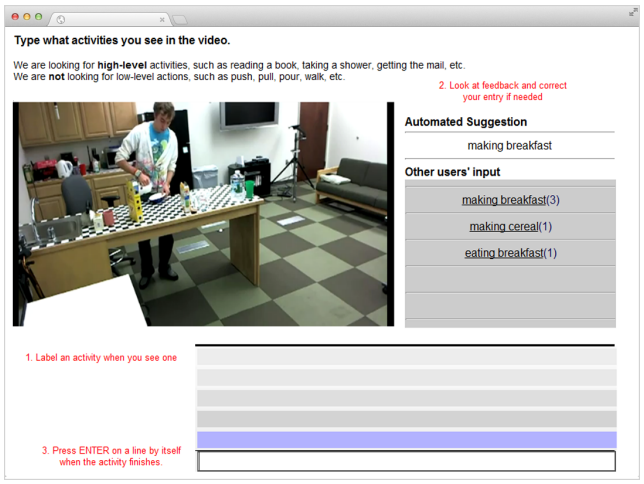

Figure 4: Legion:AR's worker interface. Workers are able to contribute new labels and vote on others. The system's HMM based activity recognition system learns over time, and suggests labels to the crowd, or when more confident, provides the label itself without human intervention.

Chorus:View

Chorus:View (Lasecki et al. 2013b) uses Chorus's conversational interaction paradigm to answer visual questions about the world around them. Conceptually, this work extends VizWiz (Bigham et al. 2010), a system for nearly real-time (within roughly 30 seconds to 1 minute) visual question answering from photographs. In VizWiz, the single-photograph requirement (that often results in difficultly for blind users when framing information), in conjunction with the delayed response time, meant that asking for a series of answers, such as when navigating in an unfamiliar area where visual cues might be the only guidance.

View lets users stream audio and video to crowd workers who are presented with a Chorus-like interface for responding to the users questions in a more natural, conversational style. While workers reply in text, a screen reader on the users phone uses text to speech conversion to make the responses accessible. View allows very rapid successions of answers to complex or multi-stage questions, and can even let workers give feedback to users on how to frame the required information. Our studies showed that blind users can get answers in a fraction of the time of VizWiz in settings where multiple questions are needed. Users were also extremely excited to get the chance to use the prototype. Our release version is in progress and will allow the thousands of VizWiz users to better access the world around them.

Legion:AR

Legion:AR (Lasecki et al. 2013a) uses the collaborative content generation that unpins Scribe to make it possible for people to supply an activity recognition system with action labels as actions occur in a video stream. Activity recognition provides automated systems with the context behind user interactions. We focus this work in monitoring settings where prompting systems can help people in their daily lives, such as helping older adults or cognitively impaired individuals live more independently, or monitoring public spaces for emergencies.

Legion:AR combines human intelligence and machine learning approaches in this labeling process. It uses a Hidden Markov Model (HMM) to learn activities from the labels provided by workers, then uses an active learning approach to ask the crowd for labels for the activities or actions the system cannot yet identify with high confidence. The system is even capable of focusing its queries on sub-activities when it can identify part of what is in the scene. When asking the crowd for labels (Figure 4), Legion:AR can also contribute its best guess. When this guess is correct it significantly reduces the time that workers must spend on their labeling task, reducing the cost and latency of the system overall. The crowd acts as a scaffold, able to answer questions reliably while scaling towards being fully automated in the future.

Adding Structure to Labels for Faster Learning Even with the ability to get assistance from the crowd when needed, training an automated system such as Legion:AR's HMM takes many examples to reliably learn a pattern. This is because the system does not understand the meaning of the events it observes, which makes it difficult to generalize this information. Unfortunately, video training data is often very costly and difficult to collect.

ARchitect (Lasecki et al. 2014b) is a system that asks simple queries to crowd workers to extract information about structural dependencies between the actions in an activity. By doing this, our experiments showed we were able to collect over 7x as much training data from each activity video, allowing automated systems to be trained faster than using label-only approaches. Eventually, my goal is to leverage human understanding to approach near one-off learning in machines by extending this type of technique.

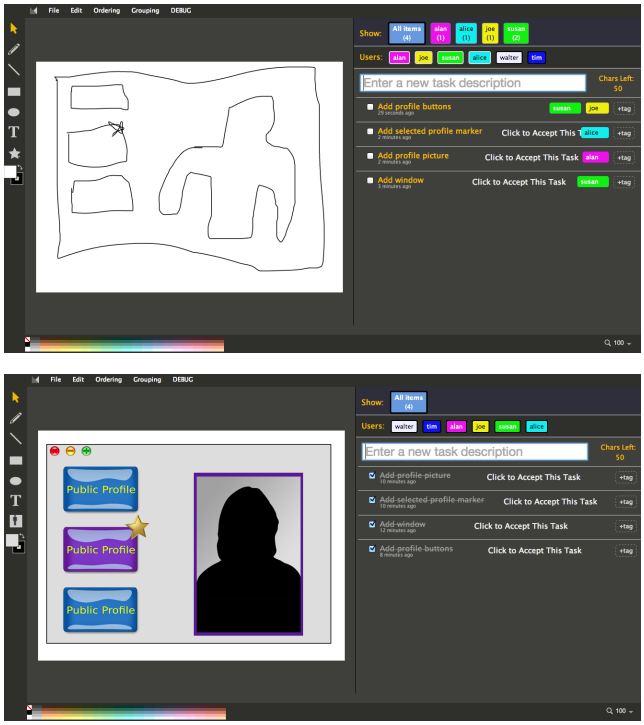

Figure 5: Apparition's web interface. Designers sketch their interface roughly while describing it in natural language (top). As they do, crowd workers collaboratively update the interface into a low-fidelity prototype (bottom). As these interfaces are created, workers provide Wizard-of-Oz control, meaning that interfaces immediately become functional, even when adding complex behaviors.

Apparition

Prototyping allows designers to quickly iterate and gather feedback, but the time it takes to create even a Wizard-of-Oz prototype restricts the effectiveness of this process (Landay and Myers 1995). In on-going work, we are creating an infrastructure and techniques for prototyping interactive systems in the time it takes to simply describe the idea visually and verbally.

Apparition is a system that uses paid crowds to make even hard-to-automate functions work immediately, allowing more fluid prototyping of interfaces that contain interactive elements or complex behaviors. As users sketch their interface and describe it aloud in natural language, crowd workers translate the sketch into traditional interface elements, add animations, and provide Wizard-of-Oz control to the prototype (Figure 5). This crowd-powered prototype can be used immediately by the design team for iteration or user study. Our approach can be combined with existing tools, and how, over time, the prototypes we develop can begin to scale towards fully implemented versions of the systems that they simulate.

Conclusions and Future Work

I create robust intelligent systems that are capable of working in real-world settings where artificial intelligence is not reliable. These systems provide a way to create new types of interactions and support new use cases, as well as providing a means of collecting training data in real situations. This would not be possible without deploying these systems, something that automated solutions cannot yet manage in the domains I study. My future work will focus on crowdsourcing workflows account for how people can *teach* machines as each tries to solve problems that arise. This lets crowd-powered systems act as a scaffold for artificial intelligence, making it possible to train intelligent systems in situ, while deployed, and then, over time, automated to improve faster, cost efficiency, and reliability.

Acknowledgements

My work has been supported by funding from the National Science Foundation, Google, Yahoo!, and a Microsoft Research Ph.D. Fellowship.

References

Bernstein, M. S.; Brandt, J.; Miller, R. C.; and Karger, D. R. 2010. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *UIST*.

Bigham, J. P.; Jayant, C.; Ji, H.; Little, G.; Miller, A.; Miller, R. C.; Miller, R.; Tatarowicz, A.; White, B.; White, S.; and Yeh, T. 2010. Vizwiz: Nearly real-time answers to visual questions. In *UIST*.

Landay, J. A., and Myers, B. A. 1995. Interactive sketching for the early stages of user interface design. In *CHI*.

Lasecki, W. S.; Murray, K. I.; White, S.; Miller, R. C.; and Bigham, J. P. 2011. Real-time crowd control of existing interfaces. In *UIST*.

Lasecki, W.; Miller, C.; Sadilek, A.; Abumoussa, A.; Borrello, D.; Kushalnagar, R.; and Bigham, J. 2012. Real-time captioning by groups of non-experts. In *UIST*.

Lasecki, W. S.; Song, Y. C.; Kautz, H.; and Bigham, J. P. 2013a. Real-time crowd labeling for deployable activity recognition. In *CSCW*.

Lasecki, W. S.; Thiha, P.; Zhong, Y.; Brady, E.; and Bigham, J. P. 2013b. Answering visual questions with conversational crowd assistants. In *ASSETS*.

Lasecki, W. S.; Wesley, R.; Nichols, J.; Kulkarni, A.; Allen, J. F.; and Bigham, J. P. 2013c. Chorus: A crowd-powered conversational assistant. In *UIST*.

Lasecki, W. S.; Gordon, M.; Koutra, D.; Jung, M.; Dow, S. P.; and Bigham, J. P. 2014a. Glance: Rapidly coding behavioral video with the crowd. In *UIST*.

Lasecki, W. S.; Weingard, L.; Ferguson, G.; and Bigham, J. P. 2014b. Finding dependencies between actions using the crowd. In *CHI*.

Lasecki, W. S.; Miller, C. D.; and Bigham, J. P. 2013. Warping time for more effective real-time crowdsourcing. In *CHI*.