Real-Time Conversational Crowd Assistants

Walter S. Lasecki
ROC HCI, Computer Science
University of Rochester
Rochester, NY 14620 USA
wlasecki@cs.rochester.edu

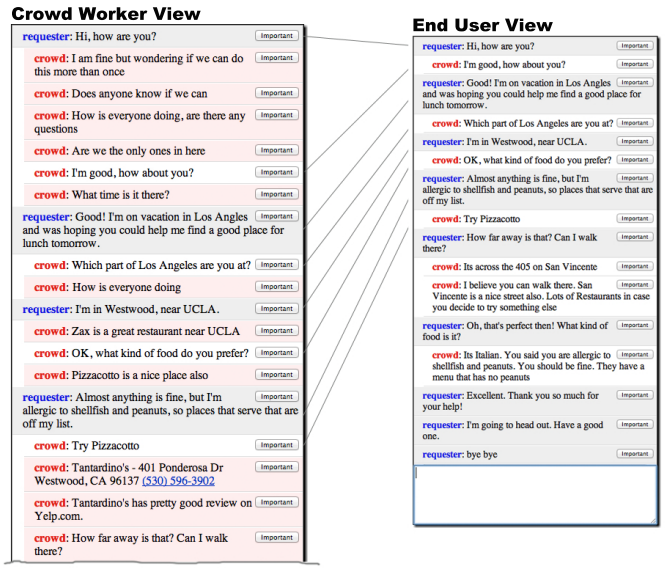

Figure 1: A conversation between a user and the crowd.

CHI'13, April 27 – May 2, 2013, Paris, France.
ACM 978-1-4503-1952-2/13/04.

Abstract

When people work together, they converse about their current actions and intentions, building a shared context to inform their collaboration. Despite decades of research attempting to replicate this natural form of interaction in computers, the capabilities of conversational assistants are still extremely limited. In this paper, we investigate how human and machine intelligence can be combined to create assistants that work even in real-world situations.

We introduce a crowd-powered conversational interface, called Chorus, that allows users to interact with a group of crowd workers as if they are a single conversional partner. We use Chorus as a personal assistant, and show that our incentive mechanism enables workers to hold consistent conversations and answer 84% of questions accurately. We then discuss a number of potential improvements that can be made by integrating artificial intelligence, and future systems that our work enables.

Author Keywords

Conversational interaction; real-time crowdsourcing; human computation; intelligent agents

ACM Classification Keywords

H.5.m [Information interfaces and presentation]: Miscellaneous.

General Terms

Human Factors, Design, Economics

Introduction

Robust conversational interfaces allow user to interact with computers more like they do with other humans. This natural interaction benefits all users, but is particularly helpful for some underrepresented groups such as older users, low-literacy users, and users with certain disabilities. In this paper, we introduce *Chorus*, a crowd-powered conversational assistant that allows groups of anonymous web workers (the crowd) to communicate with users as if they were a single, reliable individual. We present results that show Chorus' incentive mechanism is able to reliably leverage the crowd to support dialogue, answering over 84% of user queries accurately. Finally, we conclude with a discussion of future work involving the training and integration of fully automated intelligent systems that can work along side human workers, and discuss other systems that are made possible by Chorus.

Background

When people work together, they converse about their current actions and intentions, creating a shared context to inform their collaboration, thus allowing them to more easily get things done. Using natural language dialogue to interact with automated software has been a goal of artificial intelligence since the early days of computing [1]. However, despite decades of research, the complexity of human language has kept the capabilities of conversational user interfaces very limited in *(i)* the domains in which they work, *(ii)* the richness of expression they support, and *(iii)* the robustness exhibited to different users.

Real-world conversations between human partners can contain context-dependent terms or phrasing, rely on conversational memory, require commonsense knowledge about the world, events, or facts, retain memory stretching back over a long history of interactions and shared experiences, and infer meaning from incomplete and partial statements. Perhaps the most successful system to date using these methods is Apple's Siri system, and it can only handle a limited number of pre-anticipated situations. One of the weaknesses of such systems is the lack of a discourse model that can support clarification and correction dialogues in any general way.

Human Computation and Real-Time Crowdsourcing

Chorus takes a different approach to supporting dialog that blends real-time human computation with artificial intelligence to produce systems that can reliably engage in conversation. Human computation [6] has been shown to be useful in many areas that automated systems have difficulties. However, existing abstractions obtain quality work by introducing redundancy and layering into tasks, which adds time and means they are not ideally suited to interactive applications.

Recently, researchers have begun to investigate real-time human computation. VizWiz [3] introduced a queuing model to help ensure that workers were available both quickly and on demand. For Chorus to be available on demand requires multiple users to be available at the same time in order to collectively contribute. Other work has used queuing theory to recruit crowds in less than a second from existing sources of crowd workers and establish reliability bounds on using the crowd in this manner [2]. We use these recruiting systems to ensure that Chorus is available on demand.

Continuous Crowdsourcing and Crowd Agents

Traditional crowdsourcing is usually based on small discrete tasks that workers can select and complete individually. Real-time continuous crowdsourcing [4] explores using the crowd to complete tasks that would usually require workers to remain connected for longer

periods of time. These systems allow workers contribute simultaneously for as long as they choose. By intelligently combining the input of these continuous workers, the crowd can be used as a single *crowd agent*, that can outperform a single individual [4].

System

Chorus presents users with an interface to a personal assistant who is is able to hold consistent conversations, remember facts over time, and find reply with accurate responses quickly. While humans maintain natural language conversation with ease, it is often infeasible, unscalable, or expensive to hire individuals (especially experts) to act as on-demand conversational partners.

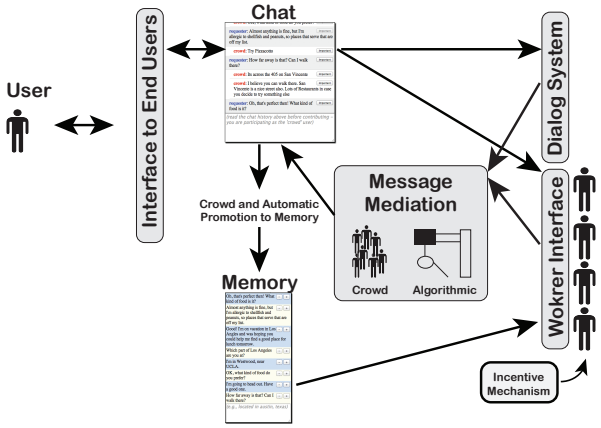

Figure 2: Chorus is a a conversational assistant that combines human and machine intelligence. To users, Chorus appears to be instant messaging client connected to a personal assistant. Behind the scenes, crowd workers (motivated by an incentive mechanism) propose and vote on responses to forward to the user. Automated dialog systems can then learn from the crowd's responses and contribute their own responses. To help maintain consistency, a working memory space is used to store important information for later.

Finding Consensus:

Using the "wisdom of the crowd" to find consistent responses is a key concept of Chorus. Workers are able to collaboratively search the web for information, building on previous suggestions until the best response is found. Prior systems have used redundancy to select among different crowd inputs [4], but this is infeasible with natural language input, which contains frequent variations. Voting allows workers to directly indicate a preferred response; however, using a voting system to find consensus requires reward schemes designed to elicit accurate answers quickly rather than rewarding sheer bulk of answers.

To encourage workers to submit only accurate responses, Chorus uses a multi-tiered reward scheme that pays workers a small amount for each interaction with the interface, a medium reward for agreeing with a response that subsequently chosen to be forwarded to the user, and a large reward for proposing a response that the crowd eventually chooses. The difference between these reward values for each of these cases can be used to adjust workers' willingness to select one option over another. For example, if the large bonus was 1000 times larger than the medium bonus, workers would have much more incentive to propose a response even if it is unlikely to be accepted since the expected reward is greater. An added challenge is to convey these rules in a way workers can understand.

To prevent participation rewards from encouraging workers to submit too many responses, we limit the number of contributions that can be said in response to a single user query. The allotted contributions are reset with each end user input. Additionally, for each allotted contribution, active workers (those who have contributed useful content recently) are paid a small amount for *not* contributing, thus removing the incentive for workers to provide responses which they are not confident in, since this would mean risking a smaller but assured reward. Since Chorus' goal is a consistent dialogue, reducing the number of responses a worker can submit to answer a single user message does not limit the system's functionality.

Relevance Rubric

On-Topic: The response furthers the topic of conversation that the user proposed by providing a relevant answer or opinion. If the relevance of a response is unclear, the user's correction or affirmation determines its status.

Off-Topic: The response is either not clearly related to the current topic (including unsolicited information about workers themselves) or provides an instantly identifiable incorrect answer to the user's question (for example, **Q**: "What is the capitol of Texas?" **A**: "Outer space.")

Clarification: The response asks a clarifying question to either confirm a fact or request the user provide more details about their question. The question must be based on the user's topic.

Generic: The response does not obviously contribute to the conversation, but is not invalid or out of place (does not break the topic flow). Generic input also does not elicit or receive a response from the user.

Experiments

We tested Chorus by holding a series of user-directed conversation with the crowd about two topics: restaurant recommendation and advice on taking care of a dog in a city apartment. A topic outline was used to guide the direction of the conversation without requiring strict adherence which would disrupt conversational flow.

We evaluated our results using a combination of quantitative and qualitative metrics. To measure the overall conversational 'quality', we calculate the total percentage of all dialogue that was forwarded to users, the portion of this dialogue that was on-topic, the error rate, the percentage of user queries that were successfully answered, the number of times workers correctly referenced facts that had been covered before they arrived, and the number of times they had to ask about such facts. Conversations were coded for each of these measures by at least two researchers according to a fixed rubric. In our initial experiments, we vary such factors as the source or size of the crowd, the incentive mechanism, or the type of automated assistance provided, but future tests will observe how these factors affect our metrics.

Our tests focused evaluating two main properties of Chorus: (i) the ability of the crowd to hold a consistent, useful conversation as a group, and (ii) the ability and willingness of the crowd to use recorded information to "remember" what happened in previous interactions with the same user, even if the workers themselves were not present. Our volunteer users were asked to follow one of two simple scripts as closely as they could in conversation. In the memory trials, we added a a 'past interaction' to the chat history and corresponding facts in the shared space window to see if workers would utilize the content.

The crowd in our trials consisted of U.S.-based Mechanical Turk workers with at least a 90% task completion rating. We had a total of 33 unique workers, with an mean of 6.7 and a median of 7 workers per trial. We limited interactions to a maximum of 10 minutes in length.

Results

Our initial tests took place over the course of two days at different times of day to increase the diversity in the crowd workers of available. A total of 371 responses were proposed by the crowd, of which, 53.1% were filtered out either in favor of another proposal or for being irrelevant. We coded each of the accepted responses for relevance to the conversation according to the rubric shown in the left margin, and ignored generic responses in calculating our results. Results for both sets of trials are shown in Table 1.

	Total Lines	Accurate Responses	Errors Made	Clarifications Asked	Questions Asked	Answers Provided	Memory Successes	Memory Failures
Consistency #1	24	9	0	0	4	4	-	-
Consistency #1	55	50	1	0	7	6	-	-
Consistency #1	33	11	0	0	2	2	-	-
Memory #1	138	53	30	3	5	3	4	2
Memory #2	63	15	1	1	4	2	1	0
Memory #3	30	29	1	1	3	3	1	0
Memory #4	28	7	0	2	3	2	2	0

Table 1: Relevance results for the conversational consistency and memory trials using Chorus.

The crowd correctly answered 84.62% of the questions user's asked during the conversations. Note that some of the questions that were *not* answered were due to the conversation moving on without resulting in a response to a given question, which occurs naturally even when communicating with a single individual. For example, when two questions are asked and the answer to one results in a new line of questioning, without first answering the other initial question.

Paying Workers for Continuous Tasks
As described before, we base worker payments on the number, accuracy, and type of contribution. This means that workers, who may remain active in the task for variable periods, are paid appropriately over any span of time, encouraging them to stay. We expect workers to get better at the jobs over time, and to receive increased pay as a result. Correspondingly, end users only need to pay for the time they use the system, not fixed minimum units of time such as an hour or a work day. For example, in initial tests with Chorus, between 7 and 50 responses from the crowd were given over the course of individual conversations, with an average of 24.86 and median of 15. Our expected required agreement (in a crowd of 5 workers) is about is 2 votes, so our task cost ranged from $0.23 to $1.66, with an average of $0.829 and median of $0.50. This is considerably cheaper than hiring an individual, dedicated employee to serve as an assistant.

Discussion and Future Work

Our results serve as a proof of concept that the underlying incentive mechanism and shared memory approach used in Chorus can facilitate consistent, helpful conversational interaction with the crowd. They also show where and how automatic intervention can be best leveraged, and points to a number of future improvements.

Improvements to Consistency
While the conversations we observed generally flowed naturally and provided useful responses, there were still a few instances of irrelevant or conflicting content being forwarded to users. In addition, requiring consensus adds delay to generating a response, taking roughly 10-30 seconds per basic response (those requiring no search) in our experiments. To help users find consensus quickly in cases where multiple valid responses are available, we have added automated support for finding semantically similar suggestions and grouping them so that workers can easily compare alternatives. Furthermore, the automated system can remove similar responses that get little support over time in order to help ensure that workers all converge to a single version of the response.

Improvements to Memory
Workers were able to recall facts from previous sessions almost every time when the content was added as pat of the trial condition. However, the list of 'important' facts resulting from worker's own curation often contained some irrelevant content such as greetings or even side-chatter amongst the workers. This indicates that the reward system for the fact highlighting task was misunderstood or needs modification. While this did not stop workers from extracting the needed information in a majority of cases, it is clear there is room for improvement.

In future version of Chorus, we will use two distinct sets of workers, each performing different tasks (conversation and memory curation) in order to focus workers more effectively on their task. To help maintain consistency between related session and preserve use privacy, we will automatically find semantic similarity between current and previous topics in order to suggest related prior facts when appropriate. This both reduced the amount of content that workers must search through to find a fact, and prevents a crowd worker contributing to any one specific session from seeing more information about the user than they need to complete their task.

A live demo of Chorus presented at UIST 2012 [5] showed the need for a definable scope of retained memories. In the demo, a single user was replaced with an ever-changing set of users, as is the case with public assistance applications such as an information kiosk in a

train station. To support this use case, we plan to use a combination of instructions to workers and natural language processing techniques such as named entity recognition to automatically detect and mark facts for further review if they may be too specific.

Training Automated Systems
An important component for crowd-powered systems is their ability to maintain the scalability of machines by learning over time to replace human intelligence. Currently, Chorus is fully operated by the crowd, but provides us the chance to learn from a *deployable* Wizard-of-Oz system. This allows automatic systems to learn from crowd interactions and eventually take over the portions of the conversation that they can do well.

To do this training, we have developed a system that allows workers to formalize each 'turn' in a conversation by filling generating a semantic frame for a specific sub-task that contains the information in the latest message. Using this formalized content, we can train multiple parts of an automated dialog system, including the parser (which converts from natural language to semantic frame), dialog manager (which determines what to do and how to respond to the user), and the natural language generator (which converts a set of information into human-readable text). Future work will focus on how even partially-trained dialog systems can work *with* the crowd to generate better conversations.

Additional Applications
Using Chorus, we also have created Legion:View, a system that provides real-time on-demand visual assistance to blind or low-vision users. View allows users to stream video to the crowd from their mobile device, then have a spoken conversation with the crowd (with the help of a screen reader) about what they see in the scene.

Conclusion

In this paper we have presented Chorus, a crowd-powered conversational assistant capable of holding natural language dialog with users. Our results show that Chorus' incentive mechanism and shared memory space can successfully allow workers to collaboratively find answers and communicate consistently with users, and remember over time. In future work, Chorus can be used in combination with automatic systems to create more robust and affordable conversational interfaces.

References

[1] Allen, J., Chambers, N., Ferguson, G., Galescu, L., Jung, H., Swift, M., and Taysom, W. Plow: a collaborative task learning agent. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, AAAI Press (2007), 1514–1519.

[2] Bernstein, M. S., Karger, D. R., Miller, R. C., and Brandt, J. R. Analytic methods for optimizing realtime crowdsourcing. In *Proceedings of Collective Intelligence*, CI 2012 (New York, NY, USA, 2012).

[3] 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. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, ACM (New York, NY, USA, 2010), 333–342.

[4] Lasecki, W., Murray, K., White, S., Miller, R. C., and Bigham, J. P. Real-time crowd control of existing interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11 (2011), 23–32.

[5] Lasecki, W. S., Wesley, R., A., K., and Bigham, J. P. Speaking with the crowd. In *In Proceedings of the Symposium on User Interface Software and Technology Demos (UIST 2012)* (2012).

[6] von Ahn, L. *Human Computation*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2005.