Speaking With the Crowd

Walter S. Lasecki, Rachel Wesley, Jeffrey P. Bigham
University of Rochester
Computer Science, ROC HCI
{wlasecki,jbigham}@cs.rochester.edu
{rwesley2}@u.rochester.edu

Anand Kulkarni
University of California, Berkeley
Department of IEOR
anandk@berkeley.edu

ABSTRACT

Automated systems are not yet able to engage in a robust dialogue with users due the complexity and ambiguity of natural language. However, humans can easily converse with one another and maintain a shared history of past interactions. In this paper, we introduce Chorus, a system that enables real-time, two-way natural language conversation between an end user and a crowd acting as a single agent. Chorus is capable of maintaining a consistent, on-topic conversation with end users across multiple sessions, despite constituent individuals perpetually joining and leaving the crowd. This is enabled by using a curated shared dialogue history.

Even though crowd members are constantly providing input, we present users with a stream of dialogue that appears to be from a single conversational partner. Experiments demonstrate that dialogue with Chorus displays elements of conversational memory and interaction consistency. Workers were able to answer 84.6% of user queries correctly, demonstrating that crowd-powered communication interfaces can serve as a robust means of interacting with software systems.

INTRODUCTION AND BACKGROUND

Interacting with automated systems using natural language dialogue has been a goal of both artificial intelligence and human-computer interaction since the early days of computing. However, the complexity of human language has made robust two-way conversations with software agents a persistent challenge [1]. Existing dialogue-based systems generally rely on a fixed input vocabularies or restricted phrasings, have a limited memory of past interactions, and use a fixed output vocabulary. 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. Even the most advanced virtual agents have difficulty handling all of these scenarios. These issues make natural language conversation an "AI-complete" problem.

UIST'12, October 7-10, 2012, Cambridge, MA, USA.

While individual people have no problem maintaining natural-language conversation, for software applications it is often infeasible, unscalable, or expensive to hire individual humans or expert agents to act as conversational partners over long periods of time. Recently, *crowd computing* has been shown to be an effective method for scalably solving problems that are beyond the capabilities of autonomous software by using groups of paid humans available via the web. Here, *the crowd* refers to a transient pool of online, semi-anonymous workers recruited for short periods of time from online microtask marketplaces such as Amazon's Mechanical Turk. Crowd computing models are able to provide software with many of the abilities of individual humans while maintaining much of the scalability of autonomous software, while presenting a new range of challenges in reliability, incentivization, and accuracy.

Many recent crowd computing applications have used the crowd to interpret natural language instructions provided by the user, in applications ranging from speech recognition [3] to interface control [5] to document editing [2]. However, this type of communication has been mostly one-directional, with the user making requests to the crowd. These applications restrict the crowd's response to being composed of a single set of options since maintaining consistent communication is inherently difficult. The pool of online agents is always changing, thus no individual worker can be relied upon to be available at a given time to respond to a query or to continuously hold a dialogue for extended periods of time. Further, individual workers may experience delays that are beyond their control, such as network bandwidth variability, that make conversation inefficient. As a result, the crowd has historically been used as a tool to interpret human instructions, rather than the foundation of a dialogue-based system.

In this paper we present Chorus, an online collaborative interface that allows users to *speak* to the crowd, engaging in a two-way natural language conversation with multiple workers as if they were a single, reliable individual person. With Chorus, users can verbally or textually submit quires to the crowd, and workers are able to produce, agree upon and submit realistic responses to user statements and queries quickly and easily using a combination of human and machine filtering. Chorus uses three components to simulate realistic conversations. First, a *collaborative reasoning* system lets workers select reasonable responses from a range of crowd-produced suggestions, filtering out responses from individual workers that do not fit the flow of the conversa-

tion and presenting users with a consistent conversational flow. Second, a *curated memory system* lets workers highlight important parts of a conversation as they emerge and presents it in a highly visible region of the interface to increase saliency. This provides a rapidly-accessible short-term memory of facts and statements that are sorted by importance, and supplements a long-term memory of interactions that remains over time. Finally, a dynamic scoring system rewards workers for collaborative interactions that support the goal of consistent conversation, such as making useful statements or highlighting facts that are used later. These features enable the crowd to collectively learn and remember information over time.

Chorus

Issuing natural language commands to individual crowd workers is relatively straight-forward (even if sometimes unreliable) since each worker can interpret the instruction and then perform an action that is consistent with what they have done in the past. However, getting a response back from the crowd requires achieving consensus on what needs to be done and said. To support this, Chorus presents each worker with the input from other workers, letting them see what has been proposed, then asks them to select the responses that they agree with. This allows workers to not only generate their own answers, but also recognize those of others. Additionally, workers are asked to select answers which are especially important for other members of the crowd to consider while working. These facts are scored using both worker input and automatic named entity recognition and ranked in a separate window for workers to see.

Once a proposed answer has sufficient crowd agreement it is "locked in", making it visible to the user. Currently, the system requires majority crowd agreement on a message in order for it to be visible to the user. The end result is that users are not overwhelmed by a flood of repetitive and possibly competing feedback or spam. Chorus can use both crowd voting and automatic agreement found by measuring the similarity between multiple submitted answers. Automatically finding agreement allows the system to forward answer with high agreement to the user without needing to wait for the crowd to vote. The Chorus system is shown in Figure 1.

In trials with the crowd, we found that Chorus was able to maintain qualitatively consistent and natural conversations between a single user and large numbers of crowd participants, remaining on-focus with single topics much as an individual user would even as individual themes and points of discussion changed. Moreover, we found that Chorus was capable of retaining meaningful long-term conversational memory across multiple sessions, even as individual users changed. Last, we showed that Chorus could be used to enable existing autonomous interfaces with dialogue capabilities. In a demonstration scenario, we experimented with using Chorus as a personable virtual assistant for search engines. We found that that users were able to accomplish a variety of search and information retrieval tasks naturally and easily through conversational interaction with the crowd. Chorus had a 84.6% success rate for correctly answered within-task queries (any input from the user that expected

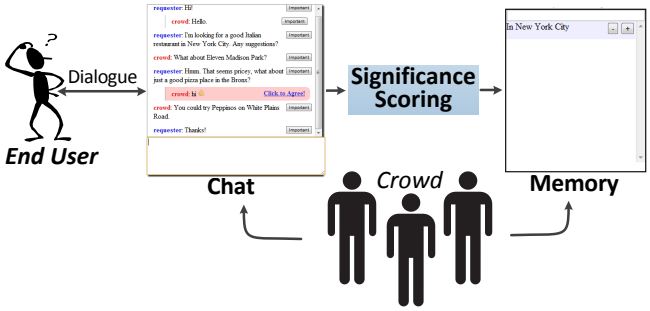

Figure 1: The Chorus system. User requests are forwarded to crowd workers, who then submit and vote on responses. Once sufficient agreement is reached, responses are made visible to users. The crowd's working memory is updated by workers selecting lines from the conversation or summarizing important facts.

a response). In cases where queries were not answered, the most typical reason was conversations flowing away from the topic (as often happens even with an individual parter), and only in rare cases was the oversight due to spam or distraction by unwanted crowd input. These findings suggest that Chorus is a robust interface for allowing disparate members of the crowd to represent a single individual during natural language conversations as an alternative to software agents. In the future, we expect Chorus will have utility as a conversational partner and as a natural-language dialogue interface to existing systems.

CONCLUSION

We have presented Chorus, a system that allows two-way dialogue between a user and the crowd. Chorus allows the crowd to have one voice, as if it were a single individual, instead of burdening the user with managing multiple lines of conversation or limiting the interaction to use only one-way communication. We demonstrated that the crowd could not only maintain a consistent conversation with users, but also learn and remember over the course of both single and repeated interactions. We then outlined how this approach could be used to more naturally interact with both existing and future crowd-powered systems.

REFERENCES

1. J. F. Allen, L. K. Schubert, G. Ferguson, P. Heeman, C. H. Hwang, T. Kato, M. Light, N. G. Martin, B. W. Miller, M. Poesio, and D. R. Traum. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *Journal of Experimental and Theoretical AI (JETAI)*, 7:7–48, 1995.
2. M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the symposium on User interface software and technology*, UIST '10, pages 313–322, 2010. ACM.
3. J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, and T. Yeh. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the symposium on User interface software and technology*, UIST '10, pages 333–342, 2010. ACM.
4. A. P. Kulkarni, M. Can, and B. Hartmann. Turkomatic: automatic recursive task and workflow design for mechanical turk. In *Proc. of CHI 2011*, pages 2053–2058, 2011.
5. W. Lasecki, K. Murray, S. White, R. C. Miller, and J. P. Bigham. Real-time crowd control of existing interfaces. In *Proceedings of the symposium on User Interface Software and Technology*, UIST '11, pages 23–32, 2011. ACM.
6. H. Zhang, E. Law, R. C. Miller, K. Z. Gajos, D. C. Parkes, and E. Horvitz. Human computation tasks with global constraints. In *Proc. of CHI 2012*, 2012. To appear.