Crowd-Powered Conversational Agents

Ting-Hao (Kenneth) Huang
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213, USA
www.cs.cmu.edu/ tinghaoh/
tinghaoh@cs.cmu.edu

Abstract

Intelligent conversational agents have quickly become a part of our digital life. However, despite decades of research, existing conversational agents are limited in scope, brittle to the complexity of language, and expensive to produce. Users today are still not capable to converse with these agents as they were more capable human dialog partners. In our work, we aim on developing the next-generation agents that have human-level capabilities of conversing. We started from the land of human computation, and explored various possibilities of incorporating automated technologies in crowd-powered systems. By deploying Chorus, a crowd-powered conversational agent, to users at scale, we are able to observe how a group of workers collectively hold a conversation, and what questions would users ask. Under the broader umbrella of conversational interactions, we explored many different forms of collaborations between workers and machines, such as having the crowd to create IF-THEN rules via conversation, or extracting entities from a running dialog in a few seconds with crowd workers. Furthermore, we proposed to incorporate automated components into Chorus to observe how technologies could improve the system. Our work demonstrated the feasibility and robustness of crowd-powered systems, and shed lights on the future of intelligent conversational agents.

Keywords crowd-powered system, conversational agent

Introduction

Intelligent conversational agents such as Apple's Siri, Amazon's Echo, and Microsoft's Cortana have quickly become a part of our digital life. There is a considerable body of research on goal-oriented spoken dialog systems ranging in domain from travel planning (Rudnicky and Xu 1999) to tutoring students (Litman and Silliman 2004). Systems vary in their approach to dialog from simple slot-filling (Bobrow et al. 1977), to complex plan-based dialog management architectures (Ferguson, Allen, and others 1998; Horvitz and Paek 1999). However, despite of decades of research, users today are still not capable to converse with these systems as they would with more capable human dialog partners. Due to the lack of fully automated methods for handling the complexity of natural language and user intent, conversational agent are largely limited to answering a small set of common queries involving topics like weather forecasts, driving directions, and similar requests.

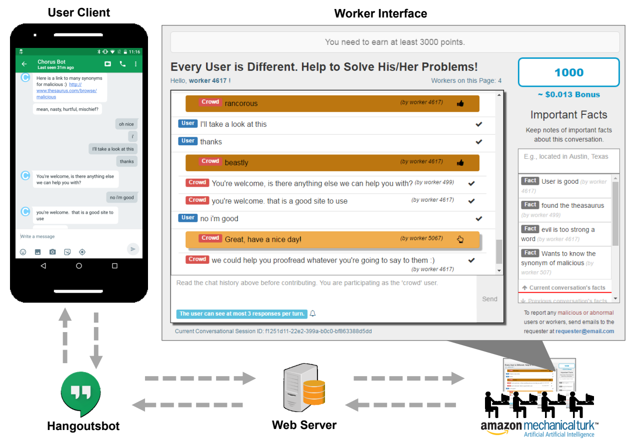

Figure 1: The Chorus UI is formed of existing Google Hangouts clients for desktop, mobile or smartwatch. Users can converse with the agent via Google Hangouts on mobile or desktop clients. Workers converse with the user via the web interface and vote on the messages suggested by other workers. Important facts can be listed so that they will be available to future workers.

In response to this situation, crowd-powered intelligent agents were proposed. Chorus is a crowd-powered conversational agent that can hold intelligent conversations about almost anything (Lasecki et al. 2012). End-users speak to it, and it responds back quickly. Chorus is powered by a dynamic group of crowd workers (recruited on-demand) who propose responses and vote the best ones through. An incentive mechanism encourages workers to contribute useful responses. Potential downsides of crowdsourcing are cost and latency (Lasecki et al. 2013). Alternatively, conversational assistants powered by trained human operators such as Magic[1] and Facebook M have also appeared in recent years.

The goal of our work is to explore the possibility of next-generation conversational agents that have human-level capabilities of conversing. We perceive crowd-powered sys-

[1]Magic: http://getmagicnow.com/

tems, which have been proven to be able to hold sophisticated conversations, as a scaffold of future conversational agents. We started from the land of human computation, and explore various ways of incorporating automated technologies in such a crowd-powered system.

Have the Crowd to Do It

We believe that observing a deployed system is capable to teach us valuable insights about the capabilities and limitations of existing technologies that lab-based study could not, especially for real-time crowd-powered agents, which were rarely deployed to real users at scale. Since our exploration starts from the land of human computation and perceives crowdsourcing as a scaffold of future systems, deploying Chorus seems inevitable.

Deployment of Chorus We developed Chorus, crowd-powered conversational assistant, and deployed it to see how users and workers would interact together when mediated by the system.[2] The system architecture and worker interface is shown in Figure 1. Chorus sophisticatedly converses with end users over time by recruiting workers on demand, which in turn decide what might be the best response for each user sentence. Up to the first month of our deployment (Huang, Lasecki, and Bigham 2016), 59 users have held conversations with Chorus during 320 conversational sessions. In this paper, we present our experience during the first month of Chorus deployment, with a focus on four challenges: *(i)* identifying when conversations are over, *(ii)* malicious users and workers, *(iii)* on-demand recruiting, and *(iv)* settings in which consensus is not enough. Our observations could assist the deployment of crowd-powered conversation systems and crowd-powered systems in general.

Machines Take Parts

Under the context of conversational interaction, we explored various possible collaborations between human workers and automated technologies. It is noteworthy that majority of our works fall within the scope of *real-time collaborations*. Crowdsourcing has been applied and known to be useful as a paradigm for data annotation in thousands of projects. However, crowd workers act collectively as an on-demand service which can interact with other automated components in real-time, was rarely studied. In this section, we structure our works under the following two categories: 1) applying automated technologies to help crowd workers in real-time, and 2) adopting real-time crowdsourcing to generate the data that can be processed by machines instantly.

Machines Assist the Crowd

Looking Forward to Automations of Chorus The currently deployed Chorus will keep running and be accessible to public for at least one year (started from May, 2016), which will be a good platform to explore various automation strategies for crowd-powered agents. We planed to utilize modern machine-learning methods to generate candidate responses and to vote for responses, as what crowd workers are required to do in the Chorus tasks. By replacing some workers with automated algorithms, we will be able to observe how automations could influence Chorus. For long term, we also plan to create a RESTful Web API server for Chorus so that not only external developers can programmatically call Chorus as a service, Chorus can also request external dialog systems for response suggestions.

[2]Chorus is available at: http://talkingtothecrowd.org/

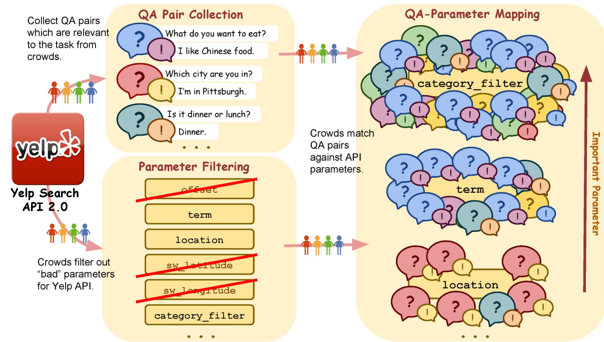

Figure 2: Offline phase of Guardian. A 3-stage Parameter Voting Workflow. Untrained crowd workers collect question and answer (QA) pairs related to the task, filter out unnatural parameters, and match each QA pair with the most relevant parameter.

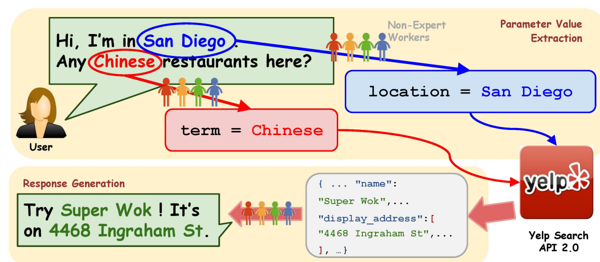

Figure 3: On-line phase of Guardian. Crowd workers extract the required parameters and turn resulting JSON into responses.

The Crowd Assists Machines

We explored various possibilities to adopt real-time crowdsourcing to generate the data that can be instantly used in the loop of an automated system, including 1) *Guardian*, a crowd-powered dialog system framework for Web APIs, 2) *Dialog ESP Game*, a method that enables a group of crowd workers annotate entities in a running dialog within a few seconds, and 3) *InstructableCrowd*, a system that creates IF-THEN rules with the crowd via conversations.

Guardian *Guardian* is a crowd-powered framework that wraps existing Web APIs into immediately usable dialog systems (Huang, Lasecki, and Bigham 2015), which contains two phases: the offline phase and the online phase.

In the offline phase of Guardian , the main goal is to connect the useful parameters in the Web APIs with actual natural language questions which are used to understand the user's query. As there are certain parameters in each Web

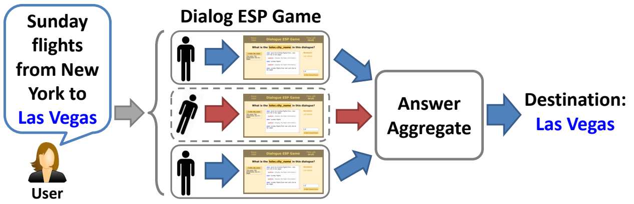

Figure 4: The crowd-powered entity extraction with a multi-player Dialog ESP Game. By aggregating input answers from all players, our approach is able to provide good quality results in seconds.

API which are more useful than others when performing an effective query on the API, it is crucial that we know which questions to ask the user to acquire the important parameters. As shown in Figure 2, there are three main steps in the offline phase, where the first two can be run concurrently. First, crowd-powered *QA pair collection* generates a set of questions (which includes follow-up questions) that will be useful in satisfying the information need of the user. Second, crowd-powered *parameter filtering* filters out "bad" parameters in the Web APIs, thus shrinking the number of candidate useful parameters for each Web API. Finally, crowd-powered *QA-parameter matching* not only matches each question with a parameter of the Web API, but also creates a ranking of which questions are more important is also acquired. This ranking enables Guardian to ask the more important questions first to faster satisfy the user's information need; In the online phase of Guardian, as shown in Figure 3, the crowd is in charge of *Dialog Management*, *Parameter Filling*, and *Response Generation*. Dialog management focuses on deciding which questions to ask the user, and when to trigger the API given the current status of the dialog. The task of parameter filling is to associate the information acquired from the user's answers with the parameters in the API. For response generation, the crowd translates the results returned by the API (which is usually in JSON format) into a natural language sentence readable by the user.

Crowd-powered Entity Extraction Modern dialog systems rely on accurate entity extraction to understand user utterances. However, entity extraction is brittle due to data scarcity, language variability, and out-of-vocabulary entities. To bridge this gap, we propose a real-time crowdsourcing solution based on the ESP game for image labeling. As shown in Figure 4, when multiple players agree, entities can be reliably extracted from an utterance. This approach is advantageous because it does not require training data. Further, it is robust to unexpected input and capable of recognizing new entities. Our approach achieves better F1-scores than that of the automated baseline for complex queries with a reasonable response time. The proposed method is also evaluated via Google Hangouts' text chat and demonstrates the feasibility of real-time crowd-powered entity extraction.

InstructableCrowd Smartphones have a wealth of sensors and effectors that can be combined to perform useful behaviors on behalf of their users. For instance, your

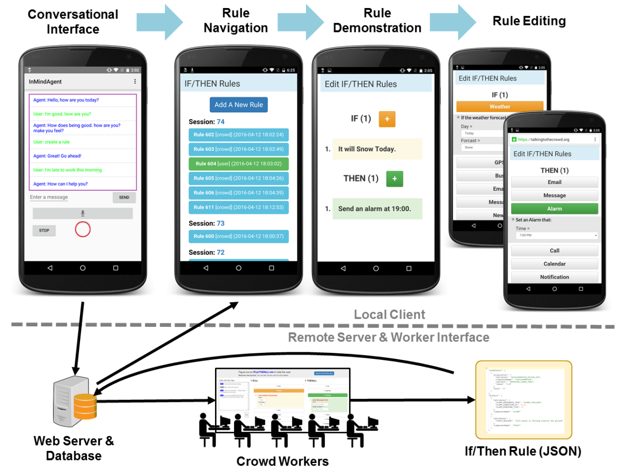

Figure 5: InstructableCrowd users have a conversation with crowd workers about a problem they are having. Crowd workers collectively create IF-THEN rules that may help the end user solve their problem using sensors and effectors available on the smartphone platform. The rules are then sent back to the user's phone for review, editing, and approval. The rules then run on the smartphone.

phone might notify you if you have an upcoming flight and aren't already headed to the airport; it might wake you up a bit earlier so you'll have time to sweep off your car if the weather forecast includes snow; or, it might text your spouse when your usual bus is running behind. We introduce InstructableCrowd (Huang, Azaria, and Bigham 2016), a system that allows end-users to collaborate with the crowd to create trigger-action (IF-THEN) rules. The system framework is shown in Figure 5. The users communicate with the crowd via voice or text message, describing a problem that they have that their phone might be able to help them solve. Crowd workers chat with the users and compose a rule with one or more IF parts connected to the user's phone sensors (*e.g.*, incoming emails, GPS location, meeting calendar, weather information, etc.), and one or more THEN parts connected to their phone's effectors (*e.g.*, sending an email, creating an alarm, posting a tweet, etc.). InstructableCrowd then sends the rules created by the crowd to the user's phone for review, so that the user can further edit and improve this rule, so that it is ready for deployment. We present the results of a user study showing that rules created by such a collaboration between crowd workers and users, are significantly more accurate than rules created solely by the users themselves.

Issues & Challenges

We sometimes received criticises about the potential lack of *technical novelty* or *scientific contributions* from reviewers, while our system contributions were often appreciated. The concept of crowd-powered system has been proposed and tested in lab-based studies. One would also perceived the idea of "having humans to do it" as very straightforward.

References

Bobrow, D. G.; Kaplan, R. M.; Kay, M.; Norman, D. A.; Thompson, H.; and Winograd, T. 1977. Gus, a frame-driven dialog system. *Artificial intelligence* 8(2):155–173.

Ferguson, G.; Allen, J. F.; et al. 1998. Trips: An integrated intelligent problem-solving assistant. In *AAAI/IAAI*, 567–572.

Horvitz, E., and Paek, T. 1999. A computational architecture for conversation. *COURSES AND LECTURES-INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES* 201–210.

Huang, T.-H. K.; Azaria, A.; and Bigham, J. P. 2016. Instructablecrowd: Creating if-then rules via conversations with the crowd. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 1555–1562. ACM.

Huang, T.-H. K.; Lasecki, W. S.; and Bigham, J. P. 2015. Guardian: A crowd-powered spoken dialog system for web apis. In *HCOMP*.

Huang, T.-H. K.; Lasecki, W. S.; and Bigham, J. P. 2016. "is there anything else i can help you with?": Challenges in deploying an on-demand crowd-powered conversational agent. In *Proceedings of Conference on Human Computation & Crowdsourcing (HCOMP 2016)*.

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 *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 23–34. ACM.

Lasecki, W. S.; Thiha, P.; Zhong, Y.; Brady, E.; and Bigham, J. P. 2013. Answering visual questions with conversational crowd assistants. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '13, 18:1–18:8. New York, NY, USA: ACM.

Litman, D. J., and Silliman, S. 2004. Itspoke: An intelligent tutoring spoken dialogue system. In *Demonstration Papers at HLT-NAACL 2004*, 5–8. Association for Computational Linguistics.

Rudnicky, A., and Xu, W. 1999. An agenda-based dialog management architecture for spoken language systems. In *IEEE Automatic Speech Recognition and Understanding Workshop*, volume 13.

Supplemental Paragraph

I plan to do my thesis proposal in November 2016, shortly after HCOMP conference. I am currently in the stage of writing my proposal, and I expect to defense my thesis at the end of August, 2017.

I expect this consortium could help me to 1) improve the organization of my thesis, and help me to understand and focus on the "theme" of my work better, 2) bring new perspectives that I was not aware of to my work, and 3) clarify the scientific contributions and technical novelty of my work.