Evorus: A Crowd-powered Conversational Assistant That Automates Itself Over Time

Ting-Hao (Kenneth) Huang **Joseph Chee Chang** **Saiganesh Swaminathan** **Jeffrey P. Bigham**
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213, USA
{tinghaoh, josephcc, saiganes, jbigham}@cs.cmu.edu

ABSTRACT

Crowd-powered conversational assistants have found to be more robust than automated systems, but do so at the cost of higher response latency and monetary costs. One promising direction is to combined the two approaches for high quality and low cost solutions. However, traditional offline approaches of building automated systems with the crowd requires first collecting training data from the crowd, and then training a model before an online system can be launched. In this paper, we introduce Evorus, a crowd-powered conversational assistant with online-learning capability that automate itself over time. Evorus expands a previous crowd-powered conversation system by reducing its reliance on the crowd over time while maintaining the robustness and reliability of human intelligence, by *(i)* allowing new chatbots to be added to help contribute possible answers, *(ii)* learning to reuse past responses to similar queries over time, and *(iii)* learning to reduce the amount of crowd oversight necessary to retain quality. Our deployment study with 28 users show that automated responses were chosen 12.84% of the time, and voting cost was reduced by 6%. Evorus introduced a new framework for constructing crowd-powered conversation systems that can gradually automate themselves using machine learning, a concept that we believe can be generalize to other types of crowd-powered systems for future research.

INTRODUCTION

Conversational assistants such as Amazon's Echo are becoming increasingly popular, but are thus far limited to speech commands in pre-determined domains. Crowd-powered assistants can be much more robust to domain and engage users in rich, multi-turn conversations. Some such systems use professional employees, such as Facebook M, whereas others use crowd workers, such as Chorus [5]. Despite their robustness, problems remain including cost, speed of responses, privacy, and quality variation [4]. In this paper, we introduce Evorus, a crowd-powered conversational assistant that is built to automate itself over time, providing a path from crowd-powered robustness to automated speed and cost. Users chat to Evorus, and it responds. The responses are chosen from suggestions offered by crowd workers and any number of automated systems. Evorus automates itself over time in three ways: *(i)* by learning to incorporate the responses of existing chatbots, *(ii)* by reusing responses previously generated by the crowd as candidates for new responses, and *(iii)* by reducing the number of votes needed for a candidate response to be sent to the user by learning to automatically detect high-quality responses. In Evorus, existing dialog systems can be incorporated by defining a REST interface on top of them that accepts information about the current conversation state, and responds with a suggested response. Evorus learns over time which automated dialog systems are most likely to be able to offer high-quality suggestions given the context of the conversation, queries them, and then those responses are forwarded to crowd workers as another suggestion. Workers then choose which of the suggested responses to use, and forwards it back to the user. When workers choose to send responses back, that is a signal that Evorus can use to automate both its selection of which dialog systems to use and to automatically pick the best responses in the future. Evorus' architecture is designed to allow future researchers to improve on its performance and the extent to which it is automated, by working on constituent problems, which are each challenging in their own right.

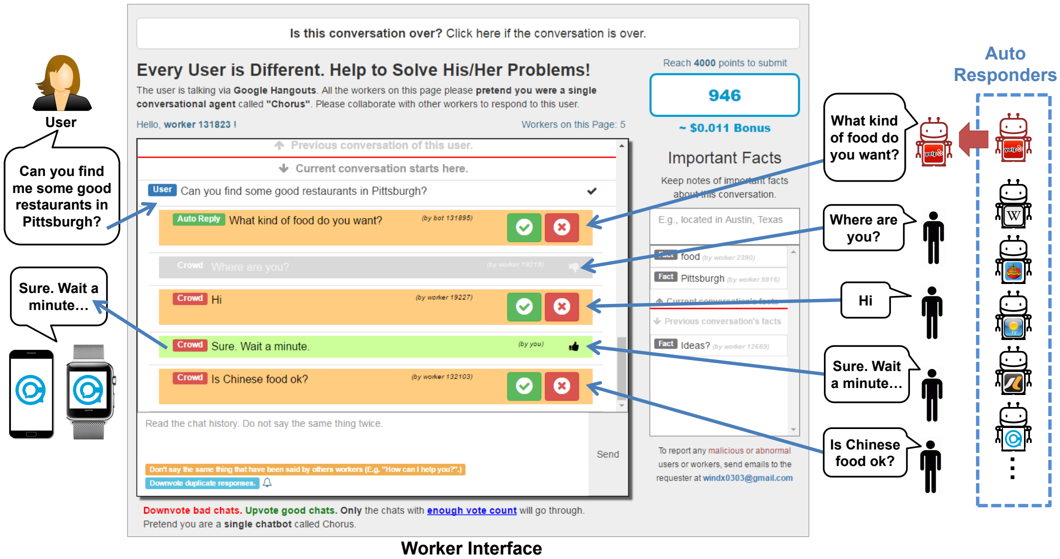
EVORUS

Evorus is a conversational assistant that is collaboratively run by real-time crowdsourcing and artificial intelligence. As shown in Figure 1, each response is first proposed by a crowd worker or an automatic chatbot, and is then selected by a collective of workers and automatic bots via a voting process.

Worker Interface: Evorus' worker interface contains two major parts (Figure 1): the *chat box* in the middle and the *fact board* on the side. In chat box crowd workers can see the messages sent by the user and the responses candidates proposed by workers and bots. Workers can click on the check mark (✔) to *upvote* on the good responses, click on the cross mark (✖) to *downvote* on the bad responses, or and type text to propose their own responses. Workers can use the fact board to keep track of important information of the current conversation. The score board on the upper right corner displays the current reward points the worker have earned in this conversation.

Upvote & Downvote: Crowd workers and bots can upvote or downvote on a response candidate. Upon calculating the voting results, we assigned a negative weight to a downvote while a upvote have a positive weights. Evorus accepts a response candidate when it accumulates sufficient vote weight.

UIST '17 Adjunct, October 22–25, 2017, Quebec City, QC, Canada.
ACM ISBN 978-1-4503-5419-6/17/10.
https://doi.org/10.1145/3131785.3131823



Is this conversation over? Click here if the conversation is over.

Every User is Different. Help to Solve His/Her Problems!

The user is talking via **Google Hangouts**. All the workers on this page please **pretend you were a single conversational agent** called **"Chorus"**. Please collaborate with other workers to respond to this user.

Hello, **worker 131823 !**

Workers on this Page: 5

Previous conversation of this user.

Current conversation starts here.

User Can you find some good restaurants in Pittsburgh? ✓

Auto Reply What kind of food do you want? (by bot 131895)

Crowd Where are you? (by worker 19219)

Crowd Hi (by worker 19227)

Crowd Sure. Wait a minute. (by you)

Crowd Is Chinese food ok? (by worker 132103)

Read the chat history. Do not say the same thing twice.

Don't say the same thing that have been said by others workers (E.g. "How can I help you?".)

Downvote duplicate responses.

Downvote bad chats. Upvote good chats. **Only** the chats with enough vote count will go through.

Pretend you are a **single chatbot** called Chorus.

Reach **4000** points to submit

946

~ $0.011 Bonus

Important Facts

Keep notes of important facts about this conversation.

E.g., located in Austin, Texas

Fact food (by worker 2390)

Fact Pittsburgh (by worker 5916)

Current conversation's facts

Previous conversation's facts

Fact Ideas? (by worker 12669)

To report any malicious or abnormal users or workers, send emails to the requester at windx0303@gmail.com

Figure 1. The Evorus worker interface shows the on-going conversation with the user, and allows workers to propose responses and up/down vote candidate responses. The up/down votes give Evorus labels to use to train its machine learning system to automatically gauge the quality of responses.

Chatbots & Vote Bots: The core concept of Evorus is to have crowd workers work with automated virtual agents (also known as "bots") on the fly to hold sophisticated conversations with users. In Evorus, we developed two types of bots: the automatic response generators, i.e., *chatbots*, and the automatic voting algorithms, *i.e.*, *vote bots*. Evorus monitored all ongoing conversations, and periodically called chatbots and vote bots to participate in active conversations. Both bots took the entire chat log as input, and based on the chat log to generate responses or vote. In Evorus, existing chatbots can be incorporated by defining a REST interface on top of them that accepts information about the current conversation state, and responses with a suggested response. When sufficient amount of chatbots were included in the bot pool, *bot selection* could potentially be an important task. Currently, we implemented four chatbots in Evorus: *Filler Bot* randomly selects one response from a set of candidates, regardless of context; *Chorus Bot* uses a retrieval-based method to find the best response from all the dialogues that deployed Chorus had; *Interview Bot* is similar to Chorus Bot just with TV interview transcript data; and the *Cleverbot* [2] is also used. We also implemented a vote bot that is powered by LibLinear [3] classifier for automatically upvoting on good responses.

DEPLOYMENT STUDY

Evorus was launched as a Google Hangouts chatbot with four chatbots for three weeks. The vote bot was included in the deployed system during the last week of deployment. 83 conversations were recorded with 28 users during around three weeks of the deployment. 639 user messages were recorded, 896 messages proposed by crowd workers were accepted, and 149 messages proposed by chatbots were accepted. During the deployment, a conversation on average contained 7.61 user messages, 10.67 accepted messages proposed by the crowd workers, and 1.77 accepted messages proposed by automatic chatbots. Thus, **automated responses were chosen 12.84% of the time**. Chatbots were able to propose reasonable responses which were then selected by the crowd. In the following example, the *Chorus Bot* recognized that the user was asking for travel suggestions and sent a proper response:

user i am going to Madison this weekend
[*A worker responded to the prior topic.*]
bot There are several interesting things to do there.

Cleverbot also answered some simple questions as follows:

user What should I make for dinner today? I need something simple and quick
bot Spaghetti.

In terms of upvotes, each accepted non-user message received 1.12 human upvotes and 0.12 automatic upvotes. In comparison, the system with only chatbots but no vote bot, each accepted non-user message received 1.19 human upvotes. **Crowd voting was thus reduced by 6%**.

CONCLUSION & FUTURE WORK

In this paper, we have introduced Evorus, a crowd-powered conversational assistant that learns to automate itself over time. We have introduced components that automatically propose responses and help to vote through responses that have been proposed. These are integrated into the crowd-powered system using an architecture that allows for increasing automation over time through learning and integration of new automated components. In our deployment, we demonstrated that already 12.84% of responses could be automated and 6% of votes could be replaced with the automated voting component. Currently, the automated bots included in Evorus are non-task chatterbots. Our next step is to include task-oriented dialog systems into the mix to allow Evorus to better handle assistance in domains in which these systems work well.

ACKNOWLEDGEMENTS

This research was supported by the Yahoo! InMind Project [1]. We also thank the workers on Mechanical Turk who participated in our experiments.

REFERENCES

1. Azaria, A., and Hong, J. Recommender system with personality. In *Proceedings of the 10th ACM conference on Recommender systems*, ACM (2016).

2. Carpenter, R. Cleverbot, 2006. [Online; accessed 08-March-2017].

3. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. Liblinear - a library for large linear classification, 2008. The Weka classifier works with version 1.33 of LIBLINEAR.

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

5. Lasecki, W. S., Wesley, R., Nichols, J., Kulkarni, A., Allen, J. F., and Bigham, J. P. Chorus: A crowd-powered conversational assistant. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST ’13, ACM (New York, NY, USA, 2013), 151–162.