Automated Support for Collective Memory of Conversational Interactions

Walter S. Lasecki[1] **and Jeffrey P. Bigham**[2,1]

ROC HCI, Computer Science[1]
University of Rochester
wlasecki@cs.rochester.edu

Human Computer Interaction Institute[2]
Carnegie Mellon University
jbigham@cmu.edu

Abstract

Maintaining consistency is a difficult challenge in crowd-powered systems in which constituent crowd workers may change over time. We discuss an initial outline for Chorus:Mnemonic, a system that augments the crowd's collective memory of a conversation by automatically recovering past knowledge based on topic, allowing the system to support consistent multi-session interactions. We present the design of the system itself, and discuss methods for testing its effectiveness. Our goal is to provide consistency between long interactions with crowd-powered conversational assistants by using AI to augment crowd workers.

Introduction

Intelligent conversational interaction with computer systems has long been a goal of both Artificial Intelligence (AI) and Human-Computer Interaction (HCI) (Allen et al. 2001). Recently, Chorus (Lasecki et al. 2013) has used the crowd to enable general natural language conversational interaction that AI is not yet capable of by using a crowd of human workers to provide input in real-time. However, long-term conversational memory of interaction with the user in past sessions still provides a difficult challenge. In this paper we discuss an initial outline for Chorus:Mnemonic, a system that augments the collective memory of the crowd with an automated system for managing and storing memories more reliably over longer periods of time. We present both the design of the system itself, as well as methods for testing its effectiveness. Our goal is to achieve consistency over longer interactions with crowd-powered conversational assistants, and show how AI can be used to augment crowd memory.

Background

Chorus (Lasecki et al. 2013) is a system that enables conversational interaction with the crowd, as if it was a single consistent individual, called a *crowd agent*. The crowd agent model was originally proposed in Legion (Lasecki et al. 2011), a system that was able to control existing user interfaces with natural language by using the crowd. To adhere to this model, Chorus must be able to not only provide accurate responses to an end-user, but also be *consistent*. We define consistency as the ability to generate non-conflicting responses and remember past information discussed in previous conversations. We focus on the latter in this paper.

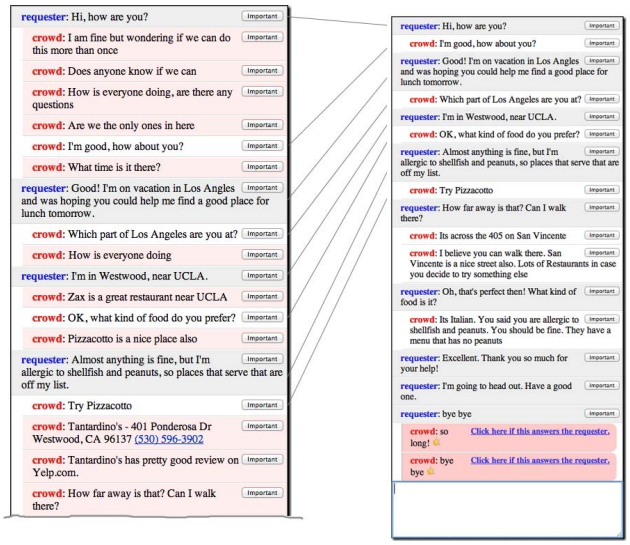

Figure 1: An example of a conversation in Chorus being held between an end-user and the crowd.

Chorus is able to generate non-conflicting responses by using an incentive mechanism that motivates workers to both propose responses and filter those generated by others. Accurate and consistent responses (the most agreed upon) are then forwarded to the end-user. Chorus supports consistency by using a memory interface designed to allow workers to curate the knowledge of the group by taking notes about important facts that have arisen from the conversation, such as the current task (e.g. "looking for a 4-star restaurant in Houston") or properties of the user themselves (e.g. "user is allergic to shellfish"). Experiments with Chorus have demonstrated workers are willing and able to use this fact list to inform their interaction with the user, even if they were not present when a fact was said, in 80% of conversations.

However, in Chorus, the memory interface only maintains 10 facts at a time, which are removed from the list when other facts become more important to the current conversation. This limits the length of the crowd's (reliable) collective memory, even if returning workers remember some facts from previous sessions (Lasecki et al. 2012). Simply making the list of recorded facts longer does not suffice for two reasons: i) workers would be required to browse longer lists, increasing the chance that they will be unwilling or unable to find a given piece of information, and ii) making all of these facts openly available to workers poses a privacy risk to users, who may divulge several individually innocu-

ous facts over time that can jointly be personally identifying. Chorus:Mnemonic focuses on using an automated system to retrieve information from a hidden database so that workers are only shown a small subset of the most relevant information about the user at any one time.

In order to understand what is being discussed by the user and crowd, we leverage research in topic modeling, such as latent Dirichlet allocation (LDA), to extract topics from the documents. While LDA approaches cannot give an exact rank of the importance of a specific topic relative to others, there are also approaches that can approximate topic rank, such as correlated topic models (Blei and Lafferty 2005).

Chorus:Mnemonic

With Chorus:Mnemonic, our goal is to merge the benefits of both machine and human intelligence to perform better then either could individually. Specifically, we plan to explore how long-term memory can be supported without burdening workers with an over-abundance of information or revealing too much (potentially sensitive) information about a user to any single individual.

Worker Interface

We begin by using a memory interface similar to the one used by Chorus. Workers can see messages in the conversation and currently posted facts. They can vote for or against a fact being relevant to the current topic of conversation. However, unlike Chorus, we make this curation of conversational history a separate task instead of using the same workers for both conversation and memory – reducing the cognitive load on workers. In initial tests we found this improved the quality of workers' contributions. We also bring workers in to complete segments of the conversational history (roughly 10-20 messages at a time), since this is more cost efficient than keeping a worker around continuously through the conversation, and an on-demand response is never needed as it is for the chat task. Furthermore, because workers can easily read the recent chat history, the memory interface only needs to act as a longer-term memory for the crowd.

Learning Relevance to Topic of Conversation

Once we obtain user input for the facts in the visible memory window, we correlate the rank of these short messages with the topic model extracted from the messages in the recent history (a short span containing the messages shown to workers). Each time workers re-rank and update the facts in memory, the weighted correlation is re-calculated. Using these correlation scores, we can begin to associate facts with topics over the course of the conversation, as facts rise to relevance, then fall from the list. Once a message is not visible on the list, it is not considered in the ranking.

By using the system's knowledge of the correlation between topics and a set of facts, facts can be reranked by the system even when they are not in the set visible to workers. This means that while workers contribute feedback to only a manageable set of at most 10 facts, the system can manage a large database of previously seen facts, showing them to workers when they are most relevant. topic of discussion.

In order to make the mapping between recorded information and topic more accurate, we require workers to write their own summaries of information in the messages they see, instead of being able to select entire messages to add to memory. This helps separate out multiple pieces of information that might be found in a single message into individual facts in memory (and workers are instructed to focus on one fact per post). To prevent repeated facts from being added to the database, the semantic similarity of facts can be compared, and if this is not enough, a binary comparison task can be sent to the crowd to determine if two inputs describe the same fact. This can be done as a relatively cheap offline task, and might be worth the small additional expense (if used sparingly) in order to consolidate facts for long-term benefit to the end user.

Conclusions and Future Work

There are a number of future directions of this work. Tests will focus first on if, after being trained on crowd feedback, the facts recalled by the system are indeed relevant to the conversations that they are visible in. We can evaluate our results using approaches from the information retrieval (IR) community. For instance, because we have at most 10 facts that can be displayed, we can use a discounted cumulative gain measurement to find the precision and recall in the top 10 (most confident) results generated by the system (Järvelin and Kekäläinen 2002). We will also investigate how workers support good suggestions and punish (down-vote) bad ones.

Another issue is the effect on privacy, we could even selectively filter the workers' ability to participate in a task involving information they have not previously seen. This "silos" workers, who will be able to participate given a fixed piece of knowledge, but without learning too much about a user over repeated interactions. We aim to measure how much information spread can be controlled in this way.

Chorus:Mnemonic presents a new approach to maintaining long-term conversational memory in crowd-powered systems, such as Chorus. It also takes steps towards future methods for using the crowd to curate memories.

Acknowledgements

This work was funded by the National Science Foundation and a Microsoft Research Ph.D. Fellowship.

References

Allen, J. F.; Byron, D. K.; Dzikovska, M.; Ferguson, G.; Galescu, L.; and Stent, A. 2001. Towards conversational human-computer interaction. *AI MAGAZINE* 22.

Blei, D. M., and Lafferty, J. D. 2005. Correlated topic models. In *NIPS 2005*.

Järvelin, K., and Kekäläinen, J. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst. 2002*.

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 2011*.

Lasecki, W. S.; White, S.; Murray, K. I.; and Bigham, J. P. 2012. Crowd memory: Learning in the collective. In *CI 2012*.

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