Finding Mnemo: Hybrid Intelligence Memory in a Crowd-Powered Dialog System

SAI R. GOURAVAJHALA, YOUXUAN JIANG, PREETRAJ KAUR, JARIR CHAAR[†], and WALTER S. LASECKI
Computer Science and Engineering, University of Michigan | [†]**IBM Research**

1. INTRODUCTION AND BACKGROUND

While dialog systems can provide a more powerful and natural way to interact with computational tools [Allen et al. 2001a; Allen et al. 2001b], robustly understanding discourse in natural language is beyond the scope of current automated approaches. By leveraging crowds of human workers to respond to end-user queries, crowd-powered dialog systems make it possible to create working systems today, while simultaneously generating data to help train future machine learning (ML) based approaches [Lasecki et al. 2013; Huang et al. 2015; Huang et al. 2016; Huang et al. 2018]. The crowds behind these systems are constantly changing, and no single worker can be relied on to be present between multiple conversational sessions. As a result, context can be lost when interactions span multiple sessions and take place over longer time scales. To provide effective replies, these crowd-powered methods need to maintain consistency over time through a shared conversational context, such as a chat history [Fono and Baecker 2006; Baecker et al. 2007]. However, having workers scroll through a long history log is costly (in terms of time) and may reduce workers' ability to participate in the conversations in real-time. Moreover, crowd-powered systems fail to map information from past dialogs to present ones, leading to a lack of contextual memory about the user and unnecessary repetition of information across conversations [Lasecki and Bigham 2013]. There is currently no way to extract concise, conversational context from those dialogs.

However, people naturally curate between-session context without thinking about it (e.g., memories about their conversations and conversational partners) and can recall information relevant to current topics with relative ease, even over long time spans [Clark et al. 1991]. Unfortunately, querying crowd workers to identify this context is difficult *because* this is a subconsious process that does not require explicit effort. But, what if we *could* tease out these latent mental models of information-saving that we innately build and instantiate this storage in our dialog systems?

We propose a methodology for note generation for conversational context maintenance by saving and aggregating human-generated notes from goal-oriented dialogs. We implement and evaluate our approach in Mnemo, a crowd-powered dialog plug-in that allows crowd workers to read dialogs and predict, curate, and save critical information into notes that will be relevant for *future* conversations, which is not a capability of existing crowdsourced summarization techniques. Our findings show that combining worker-generated notes (which would be hard or impossible to do with automatically extracting summaries) with aggregation methods that act as tunable "knobs" allow collective responses to outperform individuals' responses.

2. SYSTEM

We built the Mnemo plugin on top of Chorus [Lasecki et al. 2013]. Mnemo's interface consists of four parts (Figure 1): (1) the *DialogView* panel is where workers see the dialog history. Workers can select

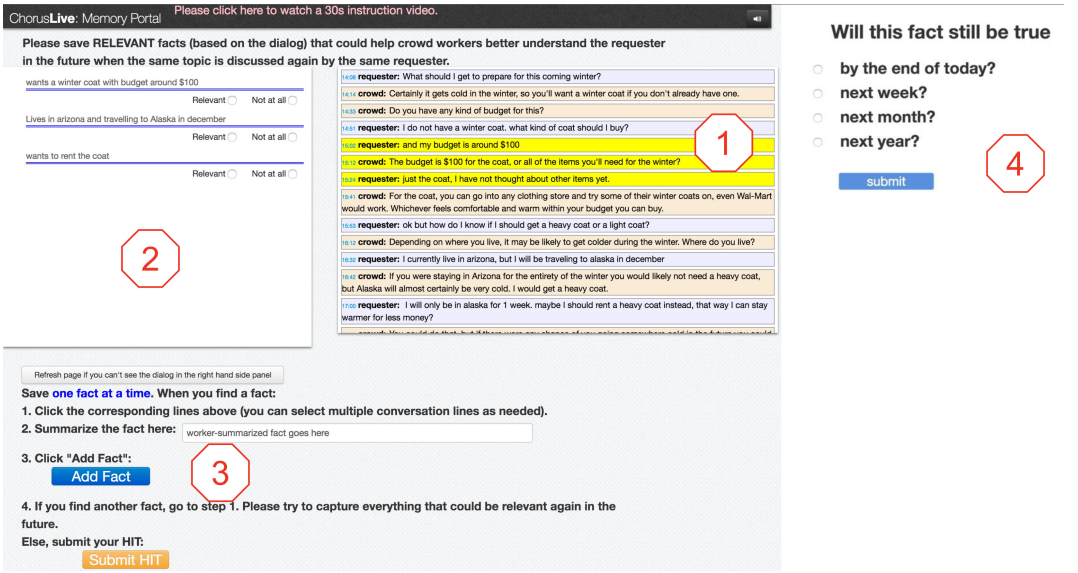

Fig. 1. Mnemo's interface consists of four parts: (1) DialogView: display raw dialog lines; (2) NoteView: display already-saved notes; (3) NoteSummary: allow workers to summarize notes; and (4) TimeSelect: allow workers to estimate a note's longevity.

one or more sentences from the dialog and group them as part of one "note"; (2) the *NoteView* panel, where worker-created notes are displayed; (3) the *NoteSummary* panel, where workers summarize relevant information in the selected lines using their own words as suggested by [Lasecki et al. 2012]. By *relevance*, we refer to notes that will still be true about the user in the long term (e.g., a requester's allergy information), rather than notes that will be true in general (e.g., information about a particular restaurant); (4) the *TimeSelect* view, which allows workers to estimate a note's period of relevance (will be true for a day, a week, a month, and a year).

3. EVALUATION

Our central question is: given a conversation about a topic and a requester's context and preferences, *how accurately can workers predict relevant notes about that user for use in future interactions?* Here, we describe how we generate dialogs, measure performance, and evaluate Mnemo's workflow.

Generating dialog histories. We recruited five student participants (none of whom had used Mnemo before) and asked them to generate ten human-human dialogs across five topics. To extract notes, ten unique workers were recruited for each dialog from Amazon Mechanical Turk. Two of the authors used Mnemo's interface to independently generate relevant facts for each of the ten dialogs, where the final gold standard was constructed by taking the union of both of their sets.

Worker performance measures. We use precision and recall to evaluate workers' ability to capture relevant notes. For our system, high precision implies that most or all of the worker-created notes are future-relevant, with minimal irrelevant notes; high recall implies that workers capture most or all of the relevant notes for that conversation.

Task description. Workers are presented with a dialog history and are instructed to identify relevant notes one at a time by clicking on the corresponding lines in the dialog, and then providing an estimated expiration date. They can select multiple lines from the dialog, then summarize those lines into their own words into a "note". Workers were instructed to only save notes that will be relevant about the requester, so any notes that are true about the world are considered irrelevant.

4. RESULTS

Workers generated a total of 500 notes over ten dialogs (average of five notes per worker, $\sigma = 2.87$). We manually annotated each worker note as either being a true positive (matches a ground truth fact) or a false positive (does not match any ground truth facts). Of the 500 worker-generated notes, 214 were classified as true positives (42.8%), and the remaining 286 (57.2 %) were considered as false positives.

Individual workers' performance. When averaged across all dialogs, individual workers' precision and recall are 44% and 42%, respectively, showing that workers can extract future-relevant notes about users. These values are consistent across dialogs (no one topic outperformed all others).

Clustering as a means to improve collective performance. Our results show any individual contributor is imperfect (as should be expected), so we aim to use groups to collectively outperform individuals. We measure the effect of aggregation on performance by calculating average precision and recall across all possible combinations of each group size from 2 to 10. This provides a more robust measure of performance that is less tied to individual team compositions. The results show that, on average, we need five workers to exceed 90% recall. This implies that each additional worker to brings in new information, leading to increased diversity of responses.

However, we see that precision does not increase with additional workers. To address this, we develop two methods that cluster notes using content similarity, one based on worker summaries and the other based on the dialog lines they selected, with three voting schemes. For a note: 1) in *any* agreement, at least two workers agree; 2) in *majority* agreement, at least half of the workers in the group agree; and, in 3) *unanimous* agreement, all workers in the group agree. When we add one more worker, we observe that precision increases past 80% compared to the 44% individual worker baseline. However, recall drops from 42% to below 10% in both cluster conditions. More generally, clustering provides "knobs" to system builders that can be used to trade off precision and recall for particular applications. Our results show that recall-focused applications can be well-supported currently (using the "any" aggregation method) based on the fact that recall is the most important piece for a first exploration of the problem space such as this paper. If an application decides precision is most important, it can instead use the "majority" or "unanimous" agreement.

Worker errors are often not true errors. Not all false positives are irrelevant for collective memory. After manually categorizing the false positives generated by crowd workers, we find that notes can be considered irrelevant for a variety of reasons. Only 5% of notes were completely wrong; 20% were missing a critical piece of context; 6% of notes were statements about the world, rather than about the requester; 42% of false positives were still relevant in the short term; 16% of notes were missing from our researcher-generated ground truth; and finally, 11% of notes contained presupposed information necessary for the notes in the ground truth. Future dialog systems, both crowd-powered and automated, can keep in mind these categories when designing for collective memory curation.

5. CONCLUSION AND FUTURE WORK

In this paper, we have introduced the idea of collective curation of future-relevant notes during conversation to maintain context between sessions. Prior work has shown that crowd-powered dialog systems can effectively hold conversations with end users, and that crowd workers can recover context from prior conversations and effectively use it to guide future interactions when presented with relevant information. Our work is the first to present a system for capturing human insight into what information will be relevant to future conversations. It demonstrates that crowd workers are able to identify notes on their own, and that performance improves significantly combining the effort of multiple workers. Mnemo's goal is to help future crowd-powered dialog systems to move beyond one-off interactions with end users, instead building a shared conversational history that allows these agents to grow their understanding of a user over the span of many conversations.

REFERENCES

James F Allen, Donna K Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. 2001a. Toward conversational human-computer interaction. *AI magazine* 22, 4 (2001), 27.

James F Allen, George Ferguson, and Amanda Stent. 2001b. An architecture for more realistic conversational systems. In *Proceedings of the 6th international conference on Intelligent user interfaces*. ACM, 1–8.

Ronald Baecker, David Fono, Lillian Blume, Christopher Collins, and Delia Couto. 2007. Webcasting made interactive: Persistent chat for text dialogue during and about learning events. *Human Interface and the Management of Information. Interacting in Information Environments* (2007), 260–268.

Herbert H Clark, Susan E Brennan, and others. 1991. Grounding in communication. *Perspectives on socially shared cognition* 13, 1991 (1991), 127–149.

David Fono and Ron Baecker. 2006. Structuring and supporting persistent chat conversations. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. ACM, 455–458.

Ting-Hao Kenneth Huang, Joseph Chee Chang, and Jeffrey P Bigham. 2018. Evorus: A Crowd-powered Conversational Assistant Built to Automate Itself Over Time. *arXiv preprint arXiv:1801.02668* (2018).

Ting-Hao Kenneth Huang, Walter S Lasecki, and Jeffrey P Bigham. 2015. Guardian: A crowd-powered spoken dialog system for web apis. In *Third AAAI conference on human computation and crowdsourcing*.

Ting-Hao Kenneth Huang, Walter S. Lasecki Lasecki, Amos Azaria Azaria, and Jeffrey P. Bigham. 2016. "Is there anything else I can help you with?": Challenges in Deploying an On-Demand Crowd-Powered Conversational Agent. In *The Fourth AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2016)*.

Walter S Lasecki and Jeffrey P Bigham. 2013. Automated Support for Collective Memory of Conversational Interactions. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Walter S Lasecki, Rachel Wesley, Jeffrey Nichols, Anand Kulkarni, James F Allen, and Jeffrey P Bigham. 2013. Chorus: a crowd-powered conversational assistant. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 151–162.

Walter S Lasecki, Samuel C White, Kyle I Murray, and Jeffrey P Bigham. 2012. Crowd memory: Learning in the collective. *arXiv preprint arXiv:1204.3678* (2012).