Powering Spoken Language Interactions With the Crowd

Walter S. Lasecki[1], Alan Ritter[2] and Jeffrey P. Bigham[3]

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

Machine Learning Department[2]
Human Computer Interaction Institute[3]
Carnegie Mellon University
aritter@cs.cmu.edu

ABSTRACT

Spoken language interfaces (SLIs) have the potential to facilitate more natural interactions between people and computers, but realizing this potential requires spoken language interfaces to not only recognize the words people say but also the meaning and intent behind them. These problems are very difficult to solve individually and each must be solved for the other to perform optimally. As a result, it is very difficult to build prototype systems that work well enough to allow spoken language interaction to be studied in the real world. This has limited our ability to understand how users would want to interact with such systems and largely prevented the collection of general high-quality training data that could improve these systems. In this paper, we present a crowd-powered pipeline that allows robust, interactive SLIs to be prototyped and deployed today. The pipeline is at first entirely powered by human intelligence, but smoothly scales towards being fully automated by using the data it collects to improve the system.

Keywords

Crowdsourcing; conversational interaction; spoken dialog systems

Categories and Subject Descriptors

H.5.m. [**Information Interfaces and Presentation**]: Misc.

1. INTRODUCTION

Spoken language interfaces provide users with a more natural way to interact with computers and information. However, in order to be truly helpful, these systems must be intelligent enough to understand both the content and context of users' speech, as well as their meaning and underlying intentions. Doing this effectively requires solving two very difficult problems. First, the system must capture speech and convert it to text. Second, the system must interpret natural language. To make the situation worse, these two problems confound one another: without understanding the meaning of the text, it is hard to accurately capture speech input; and without accurate speech input, it is hard to understand meaning. Finally, data-driven approaches are difficult to bootstrap because current systems cannot reliably hold conversations long enough to collect realistic data.

The difficulty associated with developing and training these systems has resulted in spoken language interfaces that are brittle, that only work in predefined domains, and that have high error rates. As a result, they often frustrate users and give a poor initial impression of what speech-based interaction can accomplish. Reliable prototyping of dialog systems would allow for better, more realistic, training, as well as greater insights into what types of interactions are most beneficial to users. This would allow researchers to go beyond the current user biases that have arisen and limited the scope of the assistance provided by fully automated systems.

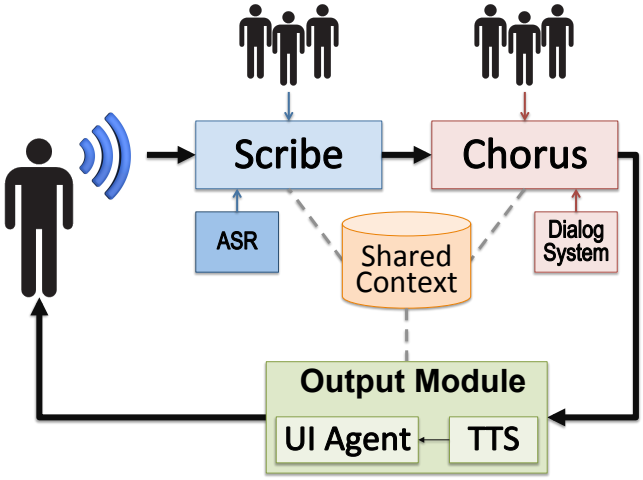

Figure 1: A crowd-powered spoken dialog system architecture. When users speak, it is converted to text in under 5 seconds by Scribe using non-expert captionists. The text is then sent to Chorus, a crowd-powered conversational assistant that generates an answer that is sent to the user in the form of text or speech (using text-to-speech), depending on the output module used. Both Scribe and Chorus can use input from existing automated systems – automatic speech recognition (ASR) and dialog systems respectively – by treating them as if they were workers providing error-prone input. Context such as topic information can be shared between speech processing, dialog, and output components to further improve performance.

In this paper, we present our vision for how crowdsourcing can be used to bring on-demand human intelligence to bear on both the speech and language processing aspects of this problem in order to allow for the collection of real-world data for training and evaluation in the absence of a fully operational automated system. We describe a pipeline (Figure 1) involving our crowd-powered speech recognition system, Scribe, and conversational assistant, Chorus, that is able to provide users with a reliable spoken language interface, while also training automated systems in ways not previously possible with existing datasets. This work also makes it possible to create working spoken language systems today that allow researchers to better understand how people interact with such systems, beyond a lab setting.

The rest of this paper is laid out as follows:

- We begin with an overview of prior work in both automated Spoken Dialog Systems (SDSs), and crowd-powered speech-to-text (Scribe) and conversational agents (Chorus).

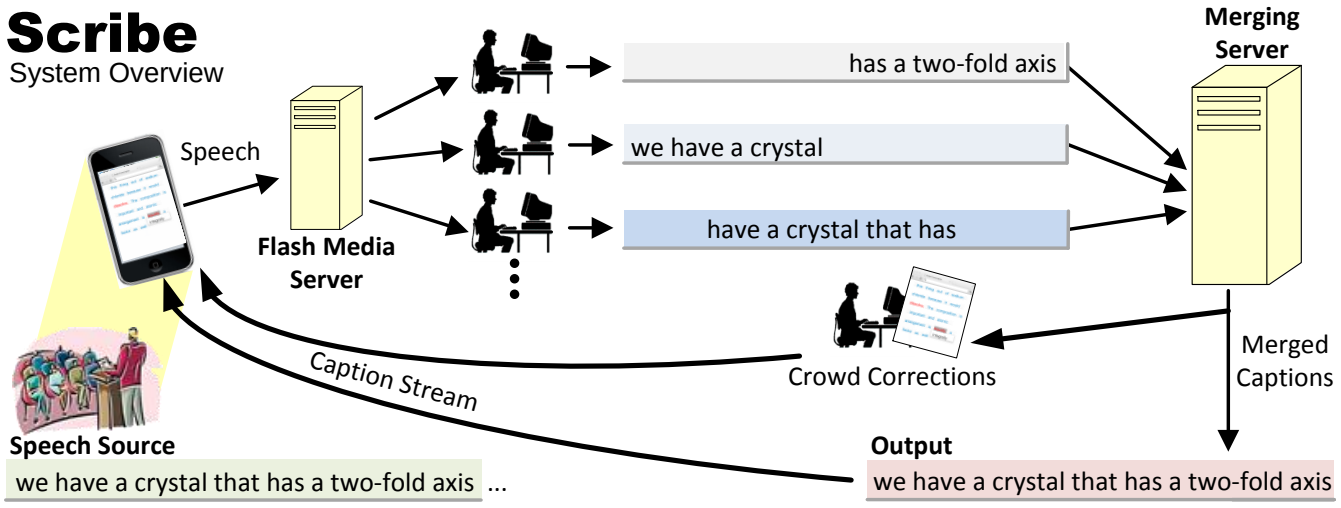

Figure 2: Scribe is a system that allows multiple non-expert workers to reliably convert speech to text in under 5 seconds by dividing the task of typing what is heard over multiple people.

- We then describe the architecture of a crowd-powered spoken dialog system that is capable of leveraging the strengths of both human and machine intelligence.
- We then discuss how crowdsourcing can be used to change the way spoken interactions are prototyped, and how automated SDSs can trained and tested, all in real-world settings.
- We conclude with a discussion of remaining challenges and future work that is suggested by our existing systems and proposed architecture.

2. RELATED WORK

We build on prior work in SDSs and crowd-powered systems.

2.1 Spoken Dialog Systems

Spoken dialogue systems have made tremendous progress over the past several years. We have a fairly good understanding of how to build dialogue systems given current technological limitations. This is evident from the availability of general purpose frameworks such as Olympus [3] which enable experts to develop systems for new domains fairly easily. These systems are highly engineered for specific scenarios, however, and quickly fail once the conversation falls outside of their scope.

To move beyond these limitations and move towards more robust dialogue systems we will need to break free of these limitations and start working on individual subproblems within the context of a working robust dialogue system. Previous work has leveraged Wizard-of-Oz studies [5] to design user interactions in the absence of a working system. Such studies are still limited because they typically use a single human "Wizard" who controls the system.

2.2 Crowd-Powered Systems

In the meantime, human computation has been shown to be an effective means of solving problems that computers cannot yet solve. By using *crowdsourcing*, open work calls to user communities or labor platforms such as Amazon's Mechanical Turk, human computation can be accessed easily and on-demand [2]. We discuss two main examples of continuous real-time systems that make the crowd appear to end-users as a single reliable entity, or *crowd agent* [9], to create interactive systems powered by the crowd.

2.2.1 *Legion: Scribe*

Legion:Scribe [7] is a system that allows groups of non-expert workers to convert speech to text in real-time (i.e., with a latency of less than 5 seconds). Scribe was initially envisioned as a means of providing more affordable real-time captions for deaf and hard of hearing students in classroom settings, where ASR is not even considered by service providers because of its poor accuracy. Instead, highly skilled professionals who charge between $100 and $300, are used. Scribe enables more affordable captions by using three to five non-expert captionists can be recruited for around $10/hour, making the total cost a fraction of that of a professional.

Scribe divides the audio input between workers (Figure 2) so that no one workers is responsible for more than they can handle. As workers caption content, their results are merged back into a single transcript by a specialized multiple-sequence alignment algorithm [13]. Scribe can also augment the audio signal (e.g., by slowing it down [8]) in order to make workers' task easier, and further improve transcription quality. More generally, Scribe provides an alternative to ASR that is more reliable, lower latency, and more adaptable to new domains. When used as part of a spoken dialog interface, Scribe can ensure reliable conversion of speech to text, which can then be processed by the dialog system.

2.2.2 *Chorus*

Chorus [12] is a crowd-powered conversational assistant (Figure 3) that allows multiple crowd workers to act as a single, reliable conversational partner capable of answering general knowledge questions. Chorus works by asking workers to propose and vote on one another's responses to the user. An incentive mechanism elicits accurate responses from workers by paying more for more impactful contributions – those proposing new answers that others agree with get rewarded more than those who help mark existing answers as being helpful, who in turn get paid more than those who do not propose or select answers that others in the crowd consider helpful. Workers are also able to make notes to other current and future workers in a "working memory" space that allows workers to see the current context of the conversation, even if they just joined. This allows the crowd to collectively stay on the same page, and appear to the end-user as a single conversational partner.

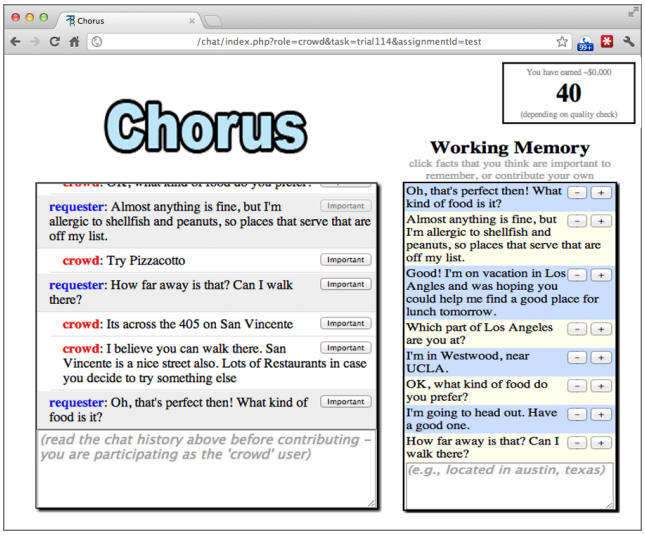

Figure 3: Chorus is a crowd-powered conversational assistant that elicits helpful, consistent answers from the crowd as if they were a single conversational partner by using an incentive mechanism and collective memory support.

By eliciting proposed responses and votes from workers in this way, we not only create a system that is able to reliably converse with users, but also generates more robust training data sets than possible with prior systems. Typically, systems are able to see a question or comment provided by a user, and (ideally) some 'accept', 'reject', or 'retry' information from the user's response. In Chorus on the other hand, each time a response is needed, multiple answers are created and voted on by the crowd, meaning alternate answers and their relative applicability rating (agreement) can be used in the training process for a dialog system.

3. CROWDSOURCING SPOKEN DIALOG INTERACTION

Using the crowd to support SDSs allows systems to make definable parts of their system robust while others remain experimentally automated, or choose to create systems that are completely reliable from the end-users' point of view, but of which only specific portions of the interaction is supported automatically.

3.1 Hybrid Systems

Our goal is to create a framework that uses the crowd to enhance the ability of automated systems, instead of replacing it entirely.

3.1.1 Architecture

Figure 1 shows the architecture of a hybrid spoken dialog system that uses both human and machine input to hold reliable conversations with users. Users can speak to the system naturally, and their speech will be converted by Scribe into text that Chorus can show to crowd workers and dialog systems. Chorus then formulates a response and sends the response to an output module. The output module can consist of plain text output, a text-to-speech (TTS) system, a virtual agent, or any combination of these (and more).

This architecture is, by design, very similar to that of a traditional dialog system: the Scribe plays the role of the speech recognizer, Chorus is the input parser, dialog manager, and natural language generator, and the output module provides feedback. The difference is that by using people as part of the underlying computational resources available, this architecture can more reliably maintain context, prevent conversations from getting derailed, and better understand users by leveraging common sense.

3.1.2 Building and Sharing Context

Using this hybrid architecture, we can build a shared context between different parts of the system to improve reliability, extending prior work on using POMDPs [14]. Introducing the crowd can also help further, for example, Scribe can create word sets that can be used to recover topic information for Chorus workers, who can also be asked to mark key words, helping to increase the accuracy of topic modeling. Chorus workers can use the topic models created from the proposed responses to create better language models that Scribe can in turn use to better predict which words that workers have typed are correct and fix mistakes where they are found. Furthermore, both Scribe and Chorus can elicit the tone and mood of the conversation from worker to ensure the response provided by the output module have the correct inflection and emphasis.

3.1.3 Training AI Systems

Using the architecture described above, we propose to gather high-quality training data within the context of a working system to train models models capable of automating components of the conversational agent in ways not previously possible. For example, Chorus naturally generates multiple competing responses suggested by workers, each of which is rated based on other workers' votes, providing a natural source of training data for an automated response ranking algorithm. We also envision crowd enabled open-domain spoken dialog systems as a natural way to enable the collection of conversationally relevant commonsense knowledge, complementary to existing large-scale knowledgebases such as NELL [4]. Additionally, both Scribe and Chorus allow automated systems to propose their own answers (which are treated as potentially unreliable just like any other worker) with the knowledge that the crowd can correct for their mistakes. This lets systems get feedback on even lower-confidence guesses without disrupting the end-user's experience from the system, letting the automated system "fail fast" to learn more quickly.

3.1.4 Combining Human and Machine Intelligence

Humans and machines are better at different things, and as such, combining their strengths can lead to better results than either can achieve independently [6]. For example, people are still better at holding conversations that feel natural, but cannot search for required information in a database as fast as a computer. Augmenting peoples' ability to respond appropriately by automatically fetching information that might be useful can help crowd-powered systems provide quicker responses without asking workers to generate answers based on insufficient information. Since information retrieval is difficult without complete understanding of the information need, this assistance will likely be error-prone, but the validity of the results in most cases will be obvious to workers (e.g., is the answer on-topic). This is an example of how computers can help in the cases where they provide useful responses, without having harmful effects when they make mistakes.

Humans and machines also make different types of errors, which can be modeled to help improve error detection and the ability to automatically rate confidence in a response. For example, when comparing ASR to non-expert human captioning in Scribe, we see that ASR is more likely to substitute an incorrect word that *sounds* the same (e.g., "Lexus" for "axis"), whereas people are more likely to substitute words that *mean* the same thing (e.g., "someone" for "somebody"). If both agree on a word on the other hand, experience has shown that it is much more likely to the a correct word.

3.2 Prototyping and Exploratory Interactions

Creating and prototyping SDSs for new situations and roles is often made difficult by low user expectations and systems unprepared to handle situations that are not known a priori. As a result, users may tend to avoid the system because they don't know where and when it will break. For example, when trying to prototype a system that allows users to conversationally control their desktop (as in Legion [9]), users are not used to having this ability at their disposal and may initially use the system only rarely. This means that there is minimal training data for a fully automated system to use, and when this early-stage prototype makes mistakes, it only makes users less likely to use it again in the future.

In these types of prototyping cases, the crowd can be used to fill in for the automated system the first few times a specific domain is encountered. This allows end users to learn the capabilities of the intended system, while also providing a means of collecting use-case and training data that is helpful during the continuing development of the automated system.

3.3 Deployable Systems

Crowdsourcing can also help systems already-deployed in real-world situations. Unknown domains in deployed systems are often a problem, similar to in the prototyping case. For instance, even a change as simple as a new domain where the context or lexicon used differs from prior examples can result in complete failure of an automated system. In these cases, our crowd-powered pipeline can provide a means of supporting deployed systems until enough training data can be provided.

Scribe and Chorus can both individually be used in this capacity by using ASR or an automated dialogue system in combination with an active learning approach that calls on the crowd only when that system's confidence is low. This means that systems can fluidly adapt to and learn new situations that have not been previously observed, and scale back to fully automated using this data. Similar approaches can be used with a full spoken dialog pipeline to immediately adapt to previously unseen domains.

Additionally, not all interactions that might have been prototyped using the crowd in the manner described above can be automated currently, or they fall in the "long tail" and are too costly to develop relative to each one's expected usage. This means a deployed system might have known deficiencies. In these cases, the crowd can be used to "fill in the gaps" until automated approaches can be developed. Because of the relatively higher cost of calling on the crowd versus an automated system (a few cents per response), system designers must carefully trade off when increased accuracy or functionality is worth the expense.

This is the use case explored by Legion:Ar [10], a system that allowed deployed activity recognition systems to fluidly adapt to and learn new actions that have not been previously observed by getting labels on-demand from the crowd, and scaling back to fully automated by using an active learning approach to decide when to query the crowd. It is also possible to use this style of support to help learn more complex knowledge about domains [11]. Similar approaches can be used with dialog systems to immediately adapt to previously unseen domains.

4. CHALLENGES AND FUTURE WORK

Future work will also explore how to get more direct training information from the crowd. For example, PLOW [1] is a system that allows users to train an automated system to do a task on the web using natural language walkthroughs. Using crowd workers to complete these tasks in real-time using Legion [9], and then training automated systems from descriptions of the actions performed would allow systems to learn offline from the crowd based on situations they've encountered in real domains.

The crowd can also be used in the final part of the system we have not yet discussed: the output module. For text-to-speech (TTS), workers can be asked to record short audio clips. For consistency, workers can be encouraged to complete multiple clips so the underlying voice changes as little as possible. Workers understand the context and can more accurately capture appropriate intonation for the content. TTS systems can even use this system to begin training on how to reply. For controlling an avatar, if one exists, workers can also provide this same feedback in the form of signaling what facial expressions or body language reactions should be used when speaking a response. Adding this final component would allow us to complete the loop of crowd support of dialog systems.

5. CONCLUSION

We have presented the architecture of a crowd-powered spoken dialog pipeline that is capable of robust interaction with users in open domains. This system can be used to quickly and flexibly take the place of a dialog system, or help train an existing dialog systems in order to scale towards being fully automated. For the first time, this makes it possible to rapidly iterate on spoken dialog interactions with real users, in real-world settings, to gain a better understanding of how spoken interaction can support their needs.

6. ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation under awards #IIS-1149709 and #IIS-1116051, and by a Microsoft Research Ph.D. Fellowship.

7. REFERENCES

[1] Allen, J. F., Chambers, N., Ferguson, G., Galescu, L., Jung, H., Swift, M., and Taysom, W. Plow: a collaborative task learning agent. In *AAAI* (2007), 1514–1519.

[2] Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., Miller, R., Tatarowicz, A., White, B., White, S., and Yeh, T. Vizwiz: nearly real-time answers to visual questions. In *UIST* (2010), 333–342.

[3] Bohus, D., Raux, A., Harris, T. K., Eskenazi, M., and Rudnicky, A. I. Olympus: an open-source framework for conversational spoken language interface research. In *NAACL workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technology* (2007).

[4] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E. J., and Mitchell, T. Toward an architecture for never-ending language learning. In *AAAI* (2010), 1306–1313.

[5] Dahlback, N., Jonsson, A., and Ahrenberg, L. Wizard of oz studies: why and how. *Knowledge-based systems* (1993).

[6] Kamar, E., Hacker, S., and Horvitz, E. Combining human and machine intelligence in large-scale crowdsourcing. In *AAMAS* (2012), 467–474.

[7] Lasecki, W. S., Miller, C., Sadilek, A., Abumoussa, A., Borrello, D., Kushalnagar, R., and Bigham, J. P. Real-time captioning by groups of non-experts. In *UIST* (2012), 23–34.

[8] Lasecki, W. S., Miller, C. D., and Bigham, J. P. Warping Time for More Effective Real-Time Crowdsourcing. In *CHI* (2013), 2033–2036.

[9] Lasecki, W. S., Murray, K. I., White, S., Miller, R. C., and Bigham, J. P. Real-time crowd control of existing interfaces. In *UIST* (2011), 23–32.

[10] Lasecki, W. S., Song, Y., Kautz, H., and Bigham, J. P. Real-Time Crowd Labeling for Deployable Activity Recognition. In *CSCW* (2013), 1203–1212.

[11] Lasecki, W. S., Weingard, L., Ferguson, G., and Bigham, J. P. Finding Dependencies Between Actions Using the Crowd. In *CHI* (2014).

[12] Lasecki, W. S., Wesley, R., Nichols, J., Kulkarni, A., and Bigham, J. P. Chorus: A crowd-powered conversational assistant. In *UIST* (2013), 151–162.

[13] Naim, I., Gildea, D., Lasecki, W. S., and Bigham, J. P. Text alignment for real-time crowd captioning. *NAACL* (2013).

[14] Williams, J. D. and Young, S. Partially Observable Markov Decision Processes for Spoken Dialog Systems. *Computer Speech and Language* (2007), 393–422.