The Need for Real-Time Crowd Generation of Task Lists from Speech

Sang Won Lee[1], **Yan Chen**[2], **Walter S. Lasecki**[1,2]
Computer Science & Engineering[1] School of Information[2]
MISC Group MISC Group
University of Michigan, Ann Arbor University of Michigan, Ann Arbor

{snaglee,yanchenm,wlasecki}@umich.edu

Introduction

Crowdsourcing has made robust speech-based interactive systems possible (Lasecki et al. 2012; 2014; 2015). Speech-based interaction provides an effective and immediate way for workers to understand requester intent via streaming audio (Lasecki et al. 2015; Lee et al. 2017), or in near real-time via audio recordings (Bigham et al. 2010; Nebeling et al. 2016; Chen et al. 2017; Zhong et al. 2015). Speech is a powerful way to describe requests because of the flexibility and context that can be easily provided (e.g., *Can you recommend a birthday present for a five year old boy? I don't know what kids like these days. I don't want to spend more than $30. It's for my nephew.*).

However, since the verbal description of a complex request can get lengthy, crowd workers may need to revisit it (given a recording), or ask the requester to repeat the description (given streaming audio). To that end, it would be beneficial for crowd workers to organize and archive complex requests for later navigation and retrieval information. In this paper, we introduce *Speech-To-Tasks*, a tool that takes audio input and generates a hierarchical list of sub-tasks.

Our *Speech-To-Tasks* system leverages crowds to segment audio into multiple parts and presents them hierarchically so that a complex task can be decomposed into a set of subtasks. The generated list of items can be used for crowd workers to self-coordinate, to avoid conflicts, as well as for requesters to monitor their progress. Further, converting speech into a structured summary itself can be useful in various settings (e.g., taking notes in meetings, summarizing a lecture, generating to-do items from a verbal request). We aim to provide a general solution for speech-based tasks, especially for near/real-time applications in crowdsourcing (Lasecki et al. 2011). In addition, findings from this work can help inform the design of intelligent conversational assistants like Apple Siri or Amazon Echo, and improve such systems to resolve more complex tasks.

In this paper, we outline the motivation and challenges in making complex verbal requests, and introduce our ongoing work with the following target contributions:

- The idea of converting complex request in speech into a set of subtasks in a crowdsourcing system.

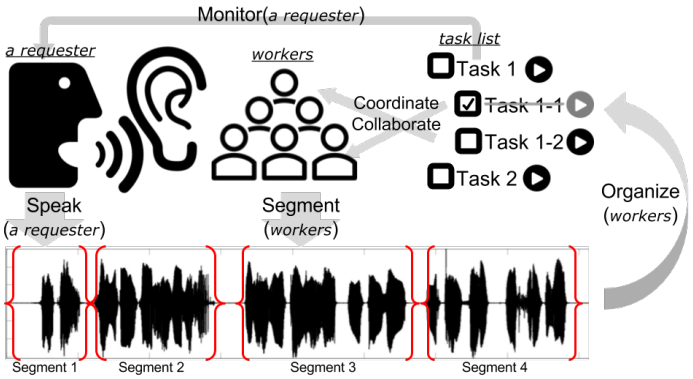

Figure 1: *Speech-To-Tasks* Process: Crowd workers segment audio into multiple parts and assign each segment to a hierarchical list. The list can be used for crowd workers to self-coordinate in case of real-time collaboration and for the requester to monitor the progress of the tasks.

- Identifying challenges and costs in verbally describing complex tasks.
- Methods and tools that help crowd workers segment speech input and synthesize a list of hierarchical subtasks.

Challenges in Describing Complex Tasks

Speech Recognition vs. Human Computation Defining tasks using text-based topic modeling or automatic speech recognition (ASR) is challenging due to error-prone results of ASR, the difficulty in understanding the intent of natural language, or missing context (Yamamoto, Ogata, and Ariki 2003; Glass et al. 2007; Repp, Grob, and Meinel 2008). Even with significant advancements in the accuracy of ASR, the consequences of errors can be costly in real-world applications and using automated methods to extract semantic relationships between audio segments remains a harder, farther-off goal. Instead, we combine human computation and simple audio processing techniques in this work in order to manually segment audio into meaningful subtasks. We then ask crowd workers to briefly annotate the audio segment or associate it with relevant elements in the task. The output of the system can be used in coordinating real-time collaboration between crowd workers once generated within the context that the list is generated.

Speech-based Interaction in Crowdsourcing Crowdsourcing is a powerful method for leveraging human intelligence that has made it possible to create systems that cannot yet be fully automated. Prior work in crowdsourcing has often recruited workers who are asked to complete small, context-free units of work called "microtasks." While microtasks are useful, only a limited range of problems can be solved using microtasks alone, and the process of decomposing a complex task into a set of microtasks is often challenging and can require special expertise.

To support a broader range of tasks, there have been efforts to make crowdsourced applications deployed and available to end users to make a request directly to crowd workers (Bigham et al. 2010; Huang et al. 2016). Some systems allow end users to make requests verbally for crowd workers to solve various tasks from writing to programming (Bigham et al. 2010; Lasecki et al. 2013; Nebeling et al. 2016; Chen et al. 2017). Instead of relying solely on ASR, these systems provide the original audio that a worker can replay if they cannot recall any part of a request. Another approach is to set-up a real-time communication channel where a worker can communicate with the requester constantly (Lasecki et al. 2015; Lee et al. 2017). In both cases, most systems assume a one-to-one relationship between the request and the response. However, in the real-world, it is more natural for people to make multiple requests in one task, as opposed to withholding one request until the previous request is complete. Furthermore, one task can be complex enough that the response needs to address multiple conditions, and the crowd workers should successfully recall all the conditions. Remembering every detail of a lengthy request is challenging, which makes workers often need to replay the entire audio snippet again or request that the end user repeat their description one more time (which adds unwanted interaction overhead). This work draws on previous works that studied tools that generate a list of tasks or decompose tasks into subtasks (Kokkalis et al. 2013; Kulkarni, Can, and Hartmann 2012).

Case Studies: Making Complex Requests Verbally Two of our previous projects exemplify the cases in which a requester needs to verbally describe a complex task to crowd workers. In SketchExpress (Lee et al. 2017), a requester and crowd workers participate in a session with real-time audio streaming, and the requester verbally describes the interactive behaviors they intend to include in the sketch prototype of graphical user interface. Crowd workers then create complex interactive behaviors of the description by a demonstrate-and-remix method introduced in the system. Typically, an interactive behavior needs to satisfy multiple different conditions to be correct. In the user study, a worker often asked the requester to repeat or to clarify the task in case they could not understand the task from the first description. In that work, we found that a requester described it 1.43 times per one interactive behavior on average ($SD = 0.62, N = 30$).

Programming is another example of a complex task with inherently interdependent sub-components. We previously explored how expert crowd workers can assist software developers on-demand (Chen, Oney, and Lasecki 2016). In the study, a software developer was given one programming task and was offered the chance to ask for help from an expert crowd worker via call. On average, developers asked for support from expert crowd workers 9.2 times during the session ($SD = 4.6, N = 12$). Expert crowd workers asked for the question to be repeated an average of 1.66 times during the session, and to clarify the request 0.42 times during the session ($SD = 0.21, SD = 0.17$, respectively, $N = 12$), even with the shared screen throughout the session. It seems that getting help from crowd workers for complex tasks comes at the price of the overhead communication between crowd workers and the requester.

In both of these case studies, a tool that can translate speech into a structured to-do list will benefit both requesters and crowd workers. It will allow requesters to describe complex tasks in one take without interruption and decomposition, and will let crowd workers navigate the audio segments to retrieve information quickly and effectively. Especially, the requesters can be less involved so that they can be handoff the requests and parallelize their work. Enabling such asynchronous collaboration can lead to a gain in productivity (Chen et al. 2017)

Structured list as a Coordination Tool Lastly, the structured of subtasks with audio segments can be used as a coordination tool for crowd workers 1) to distribute tasks and avoid conflicts and 2) to monitor if all the subtasks (requirements) are complete. The previous study used a simple to-do list to avoid the conflict when multiple workers can work on one subtask (Lasecki et al. 2015). Each item on the list can be claimed by individual workers and checked off when completed like a to-do list. The status of to-do list will provide awareness for the crowd workers and the requester to keep track of the progress of the entire task.

Ongoing Work on Speech-To-Tasks

The design and implementation of *Speech-To-Tasks* is currently in progress. The general procedure of the system is depicted in Fig. 1. The audio recording of a verbal request will be presented to crowd workers for segmentation and the sentence boundaries will be suggested to expedite the segmentation process. Once an audio-segment is created, a crowd worker can place it into an interface that can generate a to-do list that shows the hierarchy of tasks. The placed segment will create a subtask in the list, and the corresponding audio segment will be available for replay. The list will be updated as the crowd workers claim and complete them, and will be an effective indicator of the overall progress. The system will be implemented as a web-based tool that can be integrated into other crowdsourcing systems. We plan to deploy the system as part of existing crowdsourcing systems to validate if the *Speech-To-Tasks* helps crowd workers effectively complete complex tasks with less user involvement.

Acknowledgements

This project was supported by the University of Michigan as part of the MCubed 2.0 program.

References

Bigham, J. P.; Jayant, C.; Ji, H.; Little, G.; Miller, A.; Miller, R. C.; Miller, R.; Tatarowicz, A.; White, B.; White, S.; et al. 2010. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 333–342. ACM.

Chen, Y.; Lee, S. W.; Xie, Y.; Yang, Y.; Lasecki, W. S.; and Oney, S. 2017. Codeon: On-demand software development assistance. In *Proceedings of the 2017 SIGCHI Conference on Human Factors in Computing Systems*. ACM.

Chen, Y.; Oney, S.; and Lasecki, W. S. 2016. Towards providing on-demand expert support for software developers. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 3192–3203. ACM.

Glass, J.; Hazen, T. J.; Cyphers, S.; Malioutov, I.; Huynh, D.; and Barzilay, R. 2007. Recent progress in the mit spoken lecture processing project. In *Eighth Annual Conference of the International Speech Communication Association*.

Huang, T.-H. K.; Lasecki, W. S.; Azaria, A.; 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 *Fourth AAAI Conference on Human Computation and Crowdsourcing*.

Kokkalis, N.; Köhn, T.; Pfeiffer, C.; Chornyi, D.; Bernstein, M. S.; and Klemmer, S. R. 2013. Emailvalet: Managing email overload through private, accountable crowdsourcing. In *Proceedings of the 2013 conference on Computer supported cooperative work*, 1291–1300. ACM.

Kulkarni, A.; Can, M.; and Hartmann, B. 2012. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the acm 2012 conference on computer supported cooperative work*, 1003–1012. ACM.

Lasecki, W. S.; Murray, K. I.; White, S.; Miller, R. C.; and Bigham, J. P. 2011. Real-time crowd control of existing interfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, 23–32. New York, NY, USA: ACM.

Lasecki, W. S.; Miller, C. D.; Sadilek, A.; Abumoussa, A.; and Bigham, J. P. 2012. Real-time captioning by groups of non-experts. In *User interface software and technology*, UIST.

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*, 18. ACM.

Lasecki, W. S.; Gordon, M.; Koutra, D.; Jung, M. F.; Dow, S. P.; and Bigham, J. P. 2014. Glance: Rapidly coding behavioral video with the crowd. In *User Interface Software and Technology*, UIST, 1.

Lasecki, W. S.; Kim, J.; Rafter, N.; Sen, O.; Bigham, J. P.; and Bernstein, M. S. 2015. Apparition: Crowdsourced user interfaces that come to life as you sketch them. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, 1925–1934. New York, NY, USA: ACM.

Lee, S. W.; Zhang, Y.; Wong, I.; Y., Y.; OKeefe, S.; and Lasecki, W. 2017. Sketchexpress: Remixing animations for more effective crowd-powered prototyping of interactive interfaces. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, UIST. ACM.

Nebeling, M.; To, A.; Guo, A.; de Freitas, A. A.; Teevan, J.; Dow, S. P.; and Bigham, J. P. 2016. Wearwrite: Crowd-assisted writing from smartwatches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 3834–3846. ACM.

Repp, S.; Grob, A.; and Meinel, C. 2008. Browsing within lecture videos based on the chain index of speech transcription. *IEEE Transactions on learning technologies* 1(3):145–156.

Yamamoto, N.; Ogata, J.; and Ariki, Y. 2003. Topic segmentation and retrieval system for lecture videos based on spontaneous speech recognition. In *INTERSPEECH*.

Zhong, Y.; Lasecki, W. S.; Brady, E.; and Bigham, J. P. 2015. Regionspeak: Quick comprehensive spatial descriptions of complex images for blind users. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2353–2362. ACM.