Leveraging Mixed Expertise in Crowdsourcing

by

David Merritt

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2016

Doctoral Committee:

Professor Mark S. Ackerman, Chair
Assistant Professor Walter S. Lasecki
Associate Professor Mark W. Newman
Assistant Professor Sarita A. Yardi Schoenebeck

Acknowledgements

Who would have thought it was possible to combine my love for family, my fascination with astronomy, and my interest in human expertise into one (hopefully coherent) body of work? Besides God (that's a topic for a different dissertation), I'm not sure who else knew this would be the path I would take.

This work would not have been possible without my wife Beth. She gave me her time and attention when I needed it, and she let me abandon her and the kids when I needed it. Most of all, her patience with a perpetually-distracted and always-anxious husband and father is something I'm not sure I'll ever fully appreciate--but I'll try. When Ella, Addison, and Lilian read this years from now, they will have long known by then that their mother was the true champion of this effort.

I also want to thank my research "siblings" Ayse Buyuktur, Pei-Yao Hung, Elizabeth Kaziunas, and Jasmine Jones for the not-often-enough gatherings we had where you continually challenged me by the good work you were doing. I will miss the long car rides with Jasmine, when deeply insightful conversations about life, faith, and our research were the norm for a little while.

I must have had the best committee possible. I'm grateful to Mark Newman, Sarita Schoenebeck, and Walter Lasecki for serving on my committee and for giving me the luxury of having space to work in areas where I thought I should, while also prodding me in areas of this work that needed a sharper focus. I am still awed by Walter's endless

energy and passion for helping me think through my writing, even when we're minutes from a submission deadline.

Finally, with as much sincerity as I can muster in these typed words, I want to thank my advisor and committee chair, Mark Ackerman. Mark took a chance on an Air Force guy who had strings attached. He believed in me before I ever met him in person. He continued to believe in me as I struggled (repeatedly). As a matter of fact, I'm not ashamed to say that his belief in me seemed like the only thing solid enough to stand on at times. For all the hours we talked in person or on Skype, I'm still not sure if we spent more time talking about research or life. These talks sparked epiphanies about how cool qualitative analysis is, that interviewing people means I talk less, sampling (correctly) is hard, understanding probability distributions is powerful, and how having good intuitions makes system-building even more fun, but the hard part is explaining it. I'm indebted to Mark for helping me find these, and many more, nuggets of research wisdom, but I'm not convinced our talks about life didn't impact me more. We shared stories about the joys of raising kids, how unceasingly impressive our wives are, the usefulness of eidetic memory, finances, and how to enjoy (and survive) the Alaskan wilderness. I'll miss these talks with Mark, and I'm a better person and professional because of them.

Table of Contents

Acknowledgements ..... **ii**

List of Figures ..... **vi**

List of Tables ..... **viii**

List of Appendices ..... **ix**

Abstract ..... **x**

Chapter 1. Introduction **1**

1.1 Thesis Scope *3*

1.2 Research Questions *5*

1.3 Thesis Outline *6*

Chapter 2. Literature Review **8**

2.1 The Many Attributes of Expertise *9*

2.2 Using Expertise in Technical Systems *15*

2.3 Using Expertise in Crowdsourcing *24*

2.4 Summary *32*

Chapter 3. Escalier: Mixed Expertise within a Crowd **35**

3.1 Introduction and Background *36*

3.2 Use Case *38*

3.3 Escalier Formalisms *40*

3.4 Escalier Platform *42*

3.5 Implementation *48*

3.6 Evaluation Through Simulations *49*

3.7 Why Simulation *50*

3.8 Problem space and outcome metrics *52*

3.9 Model of the environment *52*

3.10 Model of user behavior *54*

3.11 Simulations *56*
3.12 Simulation Results *57*
3.13 Limitations *64*
3.14 Conclusion *64*

Chapter 4. Kurator: Mixed Expertise between Crowds **67**
4.1 Introduction *68*
4.2 Background *71*
4.3 KidKeeper Background *74*
4.4 Kurator *74*
4.5 User Study and Experiments *83*
4.6 Findings *84*
4.7 Discussion *97*
4.8 Limitations *101*
4.9 Conclusion *102*

Chapter 5. Question Finding: Mixed Subcrowds and Expertise **104**
5.1 Introduction and Background *105*
5.2 Prototype Design *106*
5.3 Evaluation *110*
5.4 Findings *111*
5.5 Discussion *125*
5.6 Issues and Limitations *130*
5.7 Conclusion *131*

Chapter 6. Conclusions **133**
6.1 General Conclusions *134*
6.2 Limitations *137*
6.3 Future Direction *138*

Appendices **140**

References **148**

List of Figures

Figure 1. The Am-I-Normal (AIN) application. AIN lets users know whether they have unusual configurations. AIN uses Escalier to obtain "conformity" data to display to the user. 39

Figure 2. How the Base Layer functions. (a) Escalier starts up with canonical (test) cases that are known to be valid (green). (b) Users report over time, growing the number of valids (green) and nearly-valids (yellow). 43

Figure 3. How Expertise Layer functions. (a) Escalier starts up with knowledge of user expertise from Q&A community data, for example, in the Expertise Layer (top). (b) Users report over time, and these reports are weighed by the expertise assessments (additional expertise data may also be added over time). 46

Figure 4. The What's-Next (WN) application. WN lets users look at what components they can add or modify, but still remain stable. 49

Figure 5. Effect of Number of Stable Configurations (300 canonical configurations; expertise disabled). The 4 lines are snapshots of number of users at that point in the simulation. 59

Figure 6. Effect of Expertise on True Positives (# of stable configurations found). This shows the effect of using Escalier's Expertise Layer's expertise assessments. 60

Figure 7. Effect of Discount Expertise Metrics on True Positives (20k stables, 300 canonicals, expertise enabled) 60

Figure 8. Kurator system diagram. Kurator starts with a collection of digital media content. A machine learning tier reduces the amount of content by filtering, based on criteria for that media type (such as no volume for audio). The crowd tier then does further refinement, producing a candidate set for the family, who is the ultimate judge for family memory. Feedback from the family can guide the improvement of the machine learning tier and the crowd tier. 69

Figure 9. Histogram of Biology and Space Expertise Scores for 120 crowd workers 113

Figure 10. Histograms of Universe, Solar System, DNA, and Stem Cell Expertise Scores for 120 crowd workers. 0.0 to 1.0 is the range of the scale that each expertise measure uses.. 115

Figure 11. Scatterplot of DNA scores versus Universe scores. 117

Figure 12. Simulation Sampling for Number of Edits to a Module..... 141

List of Tables

Table 1.Escalier simulated performance using 20k stable configurations, 300 canonical configurations, and with expertise enabled. Values are means based on 10,000 runs. 58

Table 2. ML classifier's precision, recall, and F1 scores. 86

Table 3. Crowd's precision, recall, and F1 scores. 86

Table 4. Kurator's precision, recall, and F1 scores. 86

Table 5. Curation quality, reduction in user effort, and cost savings caused by the machine learning tier of Kurator. "Album" favors recall of Definitely's, and "Best Of" favors precision for Definitely's (quality = precision for Definitely's, and %reduction = proportion of collection rated as NoWay) 92

Table 6. Correlation Matrix for Biology and Space domains as well as DNA, Stem Cells, Solar System, and Universe sub-topics (Pearson's *r* is reported). 114

Table 7. Question difficulty scores for Universe topic. Bottom, middle, and top 10% groupings are based on workers' scores on the universe expertise measure. 121

Table 8. Question difficulty scores for DNA topic. Bottom, middle, and top 10% groupings are based on workers' scores on the DNA expertise measure. 123

Table 9. Probabilistic Selection of 5 User Types (randomly selected values [0,1]) 142

Table 10. Vote Probability Based on UserType 143

Table 11. Weighted Rating Based on UserType 143

List of Appendices

Appendix A. User Search and Voting Algorithms in Escalier 141

Appendix B. Canonical and Stable Generation Algorithms in Escalier..... 144

Appendix C. Question Finding: Example Questions from the Crowd 147

Abstract

Crowdsourcing systems promise to leverage the "wisdom of crowds" to help solve many kinds of problems that are difficult to solve using only computers. Although a crowd of people inherently represents a diversity of skill levels, knowledge, and opinions, crowdsourcing system designers typically view this diversity as noise and effectively cancel it out by aggregating responses. However, we believe that by embracing crowd workers' diverse expertise levels, system designers can better leverage that knowledge to increase the wisdom of crowds.

In this thesis, we propose solutions to a limitation of current crowdsourcing approaches: not accounting for a range of expertise levels in the crowd. The current body of work in crowdsourcing does not systematically examine this, suggesting that researchers may not believe the benefits of using mixed expertise warrants the complexities of supporting it. This thesis presents two systems, Escalier and Kurator, to show that leveraging mixed expertise is a worthwhile endeavor because it materially benefits system performance, at scale, for various types of problems. We also demonstrate an effective technique, called expertise layering, to incorporate mixed expertise into crowdsourcing systems. Finally, we show that leveraging mixed expertise enables researchers to use crowdsourcing to address new types of problems.

Chapter 1. Introduction

The popularity of online crowdsourcing amongst researchers and the mainstream public has grown tremendously since the term "crowdsourcing" was first coined only ten years ago. Jeff Howe, an editor at Wired Magazine, stated:

> *"Crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call."* (Howe, 2006)

Quinn and Bederson (2011, 1405) define crowdsourcing more concisely as replacing "traditional human workers with members of the public." The idea of using an open call to members of the public to perform tasks is not new, but using the Internet to facilitate this effort has made it commonplace. In this sense, online crowdsourcing platforms like Amazon Mechanical Turk have created a new type of workforce, one made up of "anonymous"[1] workers who complete many small tasks, or micro tasks, for pay by task requesters. From a task worker's perspective, the allure of this kind of work is the flexibility in type of work and pay, as well as ultimate control over when and how much to work. From a task requester's perspective, the power of crowdsourcing is in its easy access to a large pool of human workers who can perform just about any imaginable

[1] Despite Amazon's intent for Mechanical Turkers to remain anonymous, a study by Lease et al. (2015) revealed that they are not as anonymous as was originally thought.

micro task for a reasonably small fee. While there is justifiable focus on labor issues and workplace ethics (e.g., Fort et al., 2011, Martin et al., 2014, Milland, 2014, and Teodoro et al., 2014), we focus on the crowdsourcing system designer, who ostensibly is concerned with maximizing her system's efficacy when using a crowd's inputs.

Crowdsourcing promises to help solve many kinds of problems that are difficult to solve using only computers, such as problems in natural language processing (e.g., Snow et al., 2008), computer vision (e.g., Sorokin and Forsyth 2008), and translation (e.g., Hu et al., 2011). These systems typically use an aggregation of non-expert inputs as a cheaper and faster way to replicate human expert work or to inject human expertise into an algorithmic problem. This approach has been useful, but it is missing the benefits of differentially mining what people can do, which we see as division of labor in society. Crowdsourcing system designers almost universally categorize crowd workers into expert and non-expert classes, but we believe there is more value to be gained from a finer-grained resolution of the mix of expertise. In other words, if system designers could know the continuum of expertise brought to bear by a workforce, then what could it enable?

In this thesis, we propose solutions to a limitation of current crowdsourcing approaches: not accounting for a range of expertise levels in the crowd. The current body of work in crowdsourcing suggest that researchers do not believe the benefits of using mixed expertise warrants the complexities of supporting it. This thesis demonstrates that leveraging mixed expertise is a worthwhile endeavor for crowdsourcing system designers, and that using the principles of expertise layering, discussed below, is an effective way to incorporate mixed expertise.

1.1 Thesis Scope

This thesis establishes that there are benefits to incorporating mixed expertise into crowdsourcing system design. These benefits are not limited to the system's performance, however. We believe that looking for problems where mixed expertise would be beneficial has shed light on understudied problem areas in crowdsourcing research. The work in this thesis heeds the call of Bigham et al. (2015, 12): "*Crowdsourcing has traditionally worked best, although not exclusively, for problems that required little expertise. A challenge going forward is to push on the scope of problems possible to solve with crowdsourcing by engaging with expert crowds, embedding needed expertise in the tools non-expert crowds use, or by using a flexible combination of the two*."

In short, instead of building a system to target only one level of expertise, a designer can build for a range of expertise in worker inputs, especially where the problem requires it. With this in mind, the "scope of problems possible" that we address in this thesis includes:

1) problems where the diverse expertise of a population *must* be leveraged to uncover more of a solution space,
2) and personalized, subjective problems where there are multiple valid solutions to an end user,

These two problem areas are particularly interesting because they are known difficult problems that likely would benefit from mixed-expertise, crowd-powered systems. As an example of the first type of problem, suppose we want to generate multiple versions of a textual summary of a scientific article about the solar system,

where the versions ranged from high school-level vocabulary to an expert-level one. This problem could benefit from a crowdsourcing system that recognizes a diversity of expertise people have about the solar system. A user with expert-level knowledge might be able to write summaries for other experts, and those with less expertise can write for others with similar expertise. In short, for this first type of problem, if we were to focus on obtaining only expert input, then we would be missing part of the solution space (e.g., textual summaries for non-experts).

As an example of the second type of problem, suppose a mother wants to find 20 sentimental photos of her children from her collection of 10,000 digital pictures. She may not be as concerned with finding the "best" photo(s) as she is with finding sufficiently sentimental ones. In this scenario, there are likely to be many valid solution sets. Using a crowd to find sentimental pictures would require some notion of expertise, where expertise in this problem is manifested as one's ability to predict the mother's opinion of what sentimental is. By definition, the mother has the most expertise because her opinion is the gold standard. Presumably, people who know the kinds of memories the mother likes to keep, like close family or her spouse, would have more expertise in this task than the general public. Thus the family might select solution sets similar to the mother's. On the other hand, a crowd worker who has grown children may have substantial expertise in this domain, and she may produce a sufficiently acceptable solution set different from the mother's. Using a mix of crowds, in this case a generic crowd and an expert crowd consisting of the mother and her family, could produce a more diverse range of valid solutions than if only the mother's or her family's inputs were used.

To address these types of problems, we built systems using a certain approach that aided us when making design choices. This approach, which we call *expertise layering,* is a set of principles that can be used as a technique for designing crowdsourcing systems for the problems within the scope of this thesis. Expertise layering follows two principles:

- adding expertise, through expertise assessments, should add value to the system when available, but the system should not be *required* to use expertise (i.e., the use of expertise is non-blocking), and
- expertise use should be *modular*, where different mechanisms for assessing and using expertise can be added or swapped out.

In this thesis, we introduce two systems and a study that use expertise layering, each highlighting different domains, types of crowds and expertise, and the resulting challenges.

1.2 Research Questions

The research questions this thesis answers are:

- **RQ1**. Under what conditions, and to what extent, does using mixed expertise within a crowd materially benefit a crowdsourcing system at scale?
- **RQ2**. Under what conditions, and to what extent, is there benefit when using a mix of crowds, differentiated by types and levels of expertise, to solve a problem when the crowds work on similar tasks?
- **RQ3**. Under what conditions, and to what extent, is there benefit when using a mix of subcrowds within a crowd, differentiated by expertise?

The methodology we use to answer these questions is building and studying crowd-powered systems. We use a combination of empirical evaluations as well as a large-scale simulation to assess feasibility at scale. This thesis presents two systems and a study that build on one another. Answering **RQ1** will determine whether using mixed expertise is a worthwhile endeavor for crowdsourcing system designers. The first system, Escalier, shows us there is material benefit, in theory, when using a mix of expertise within a crowd. Answering **RQ2** tells us if a mix of expertise between crowds, not just within a crowd, would still provide benefit. The second system, Kurator, builds on the first one by demonstrating empirically a problem that can be solved by using multiple crowds. Answering **RQ3** will determine if there is benefit to using subcrowds, which are groups from the same crowd but differentiated by their topic-specific expertise. A study on Question Finding uses what we found with Escalier (mixed expertise within a crowd) and Kurator (mixed expertise between crowds) to demonstrate the benefits of using multiple subcrowds, differentiated by expertise.

1.3 Thesis Outline

In this chapter, we provided initial motivation for the need to investigate ways to build crowdsourcing systems with mixed expertise in mind. The work presented in this thesis is built using the principles of expertise layering, and scoped to address the two problem areas we discussed. The rest of the thesis is framed to answer the three research questions in the context of these problems areas.

In the second chapter, we explore how human expertise has been defined and studied in order to identify the portions of the space that we can carry into the crowdsourcing literature. We follow with a review of the prior literature where technical

systems concerned with human expertise have been used to find and share this expertise. We apply the lessons learned from that review to the crowdsourcing literature. After reviewing the relevant crowdsourcing literature where expertise has been a focal point, we identify shortcomings that this thesis seeks to address.

In the third chapter, we present the Escalier system, which is intended to answer the first research question in the context of the first type of problem. We implemented a large-scale simulation to examine the feasibility of leveraging mixed expertise at scale. This work used some feedback loops and mixed expertise leveraged from the same crowd.

In the fourth chapter, we present the Kurator system in a deeper investigation into feedback loops and mixed expertise between crowds. Our work with Kurator is intended to answer the second research question in the context of the second type of problem. We evaluate the system and its components through a user study and lab experiments. This work also shed some light on the importance of identifying specialized crowds.

In the fifth chapter, we present a study on Question Finding, which we use to investigate more fully how we might identify *and* leverage specialized crowds, particularly subcrowds within a crowd, differentiated by expertise. The study on Question Finding is intended to answer the third research question in the context of both types of problems.

In the sixth chapter, we conclude with a discussion of the contributions and impact of the work in this thesis, as well as of future research directions enabled by this work.

Chapter 2. Literature Review

In crowdsourcing systems, the expertise of a crowd worker is largely assumed by the selection of the particular crowd to use. Often, experts are available for a particular problem, but a non-expert crowd is used as a cheaper, and sometimes faster, alternative. On the other hand, organizational work does not have a notion of non-experts in the workforce (Treem and Leonardi, 2016b). The predominant view of crowdsourcing researchers that expertise can be categorized into expert and non-expert workers is an oversimplification of expertise. It does not systematically account for the continuum of expertise within a crowd, even though there are material benefits to doing so, which is a focus of this thesis.

Why do we believe there is a mix of expertise in the crowd, and why do we believe this is important to crowdsourcing systems research? Our beliefs are based on the prior literature on the study of expertise, on the study of technical systems that identify and use expertise in organizations and communities, and on the study of crowdsourcing systems that have emphasized some notion of expertise.

In this chapter, we first discuss the relevant concepts from the broad literature on the study of human experts and expertise. This literature teaches us that expertise is not only widely studied, but there are many attributes of expertise and frameworks for reasoning about expertise that have yet to make their way into crowdsourcing research.

Second, we review the literature where expertise has been used in technical systems. Since this thesis is concerned with expertise in crowdsourcing *systems*, it is prudent to review the literature on technical systems that have identified and used expertise in real-world settings. This topic is widely studied in CSCW, particularly in the context of expertise finder systems. This literature teaches us the importance of the availability of data to measure expertise as well as the software architectures that are effective in practice.

Finally, we review the literature on crowdsourcing systems that have explicitly acknowledged a reliance on experts or expertise. In this literature, it is clear that expertise has been studied, but not explicitly framed and investigated as systematically as it has been in the CSCW literature. The prior work in crowdsourcing teaches us that crowdsourcing system designers typically view expertise as a binary categorization of experts and non-experts, which we believe could be to the detriment of some crowdsourcing systems.

We conclude the chapter with a summary of the takeaways from the expertise literature and the expertise finder systems literature that fill in some gaps in the crowdsourcing literature.

2.1 The Many Attributes of Expertise

The prior literature on the study of expertise reveals that there are many attributes to expertise. These attributes vary widely based on, for example, psychological or philosophical approaches to the study of expertise, methods for studying the structure of expertise, methods for studying the acquisition and maintenance of expertise, and how expertise is studied in different domains (Ericsson et al., 2006b). A complete review is

beyond the scope of this thesis, and the reader is referred to *The Cambridge Handbook of Expertise and Expert Performance* (Ericsson et al., 2006b) for detailed investigations of experts and expertise research, and *Expertise, Communication, and Organizing* (Treem and Leonardi, 2016a) for a more recent treatment on the social, communicative, and organizational aspects of expertise. Instead, this section focuses on specific attributes of expertise to clearly indicate how the work in this thesis fits together, and also where this work fits into research on expertise.

In this section, we discuss what "expertise" means, that there are many levels of expertise, the types of expertise with which we are concerned, and how this all relates to the focus of this thesis.

2.1.1 Expertise Defined

In this thesis, we adopt Ericsson's (2006a, 3) definition of expertise, which is "the characteristics, skills, and knowledge that distinguish experts from novices and less experienced people." Ericsson, among others, applies this definition to numerous domains where expertise is studied (e.g., law, medical, music, arts, and sports). Treem and Leonardi (2016b) also suggest that expertise is not objective or stable across contexts; a discussion of expertise must be in a domain-specific context.

Perhaps because of the many domains and topics of interest to expertise researchers, there are many frameworks for explaining the process of acquiring the "characteristics, skills, and knowledge" required for expert performance. As discussed in Ericsson's survey (2006a), these frameworks differ based on how researchers account for individuals' mental capacities, experience, mental organization of knowledge, learning environments, or task-based performance.

The work presented in this thesis operates under the last framework: expertise as reliably superior performance on representative tasks. This framework argues for finding representative tasks for measuring performance, where the tasks are under standardized conditions. As explained by Ericsson (2006a, 11), these standardized conditions "make it possible to measure and compare the performance of less-skilled individuals on the same tasks." These conditions are prevalent in crowdsourcing research. Thus this framework is a natural fit for the work in this thesis. Just as important, this framework acknowledges a gradient of expertise, which we discuss next.

2.1.2 Levels of Expertise

When studying the characteristics of experts, researchers typically use one of two approaches: the "absolute" approach, or the "relative" approach (Chi, 2006). In the absolute approach, the truly elite experts are studied to understand their superior performance within their respective domains. There is also a tacit assumption, in the literature, that true experts have "greater minds", "greater memory capacity", and "unique innate talent" that are domain-general (as opposed to domain-specific), which leads to domain-specific expert performance (Chi, 2006).

An alternative approach is aptly named the "relative" approach. Chi (2006, 21) explains this approach "assumes that expertise is a level of proficiency that novices can achieve." This approach acknowledges there are multiple levels of expertise, and expertise is more than a simple distinction of "novice" versus "expert." Hoffman's (1998) continuum of development, based on craft guilds of the Middle Ages, is indicative of this relative approach, where a "naive" person is someone who is ignorant of a domain, followed by novice, initiate, apprentice, journeyman, expert, and master, in increasing

order of skill or demonstrated expertise. As another example of the explicit acknowledgement of a continuum of expertise, Dreyfus and Dreyfus (1980) proposed a five-stage model of adult skill acquisition: novice, advanced beginner, competence, proficiency, and expertise. As a final example of an argument for a continuum of expertise, Collins (2016) explains that expertise is not just for esoteric contexts, but there are degrees of expertise within one domain. An example Collins offers is driving a car. This activity is nearly ubiquitous in many industrialized societies, which means it is not esoteric. However, professional car racing, which is arguably in the same broad domain, is esoteric. Thus it seems reasonable to treat expertise as a continuum instead of as an absolute level only for the elite.

In this thesis, we embrace the relative approach by taking the holistic view of expertise as being more than a binary assignment of "expert" and "novice" to individuals. As discussed below, this approach is under-utilized in crowdsourcing research, and we show in this thesis that leveraging the continuum of expertise represented in the crowd can materially benefit crowdsourcing systems.

2.1.3 Types of Expertise

Beyond the use of multiple levels of expertise, the work presented in this thesis also investigates different types of expertise. Note the concept of a continuum of expertise refers specifically to the skill or knowledge that distinguishes individuals with differing levels of expertise. By "types" of expertise, we are referring to abstractions in the criteria used to measure expertise. In particular, we are interested in both fact-based and opinion-based measurements of expertise.

Much of the study of expertise is focused on domains where expertise is measured using fact-based criteria, ostensibly because it is tractable to quantify and compare. For example, Chase and Simon (1973) studied chess players, and Larkin et al. (1980) studied physics experts, which are domains consisting of agreed-upon solutions and formal logic. Less well studied are domains where opinion-based criteria are used to measure expertise. Opinion-based criteria are evident in domains having subjective aspects for quality judgment, like dance (Noice and Noice, 2006), music (Lehmann and Gruber, 2006), and history (Voss and Wiley, 2006).

Voss and Wiley's (2006) characterization of "well-structured" and "ill-structured" domains is helpful in explaining where fact-based and opinion-based criteria apply. Voss and Wiley's well-structured domains consist of problems having single answers with agreed-upon solutions, typically using mathematics or formal logic. In other words, expertise measured using fact-based criteria is suitable for well-structured domains. Ill-structured domains contain problems with more than one answer, no agreed-upon solution, and little opportunity for formal logic or mathematics to apply. Opinion-based criteria are more suitable than fact-based criteria for ill-structured domains.

We also pose that fact-based and opinion-based criteria are the opposite ends of a spectrum of measurement criteria. Towards the fact-based end of the continuum are criteria based on, for example, mathematics, formal logic, and scientific and technical knowledge. Towards the opinion-based end of the spectrum are ill-defined criteria based on, for example, cultural understandings and personal preference. For this fact-opinion spectrum to make sense, the middle of the spectrum should represent contexts where a mix of the two criteria is used. Indeed, there are examples of this, such as judging

gymnastics, which requires certain compulsory skills and techniques to be demonstrated, but there is also subjective scoring, like artistry, that is calculated into overall scoring.

Although this fact-opinion spectrum represents many types of expertise, this thesis focuses on three points on the spectrum, which we discuss next.

2.1.4 Tying it All Together

As mentioned, we anchor the definition of expertise to the idea of it being domain-specific. Expertise in one domain may not correlate to expertise in other domains. As well, a person may have a high level of expertise in many domains while simultaneously having little expertise in others. Assuming no two people are perfectly alike, a crowd of people inherently represents a diversity of skill levels, knowledge, and opinions, applied to various domains. Typically, crowdsourcing system designers view this diversity as noise, and they effectively cancel it out by aggregating responses over a large enough *n*. The aggregated response is treated as "the wisdom of crowds" (Surowiecki, 2005). However, our hope in this thesis is to embrace crowd workers' diverse expertise levels. This diversity is what we mean when we use the term "mixed expertise" throughout this thesis. We want to tease out the differences in expertise within that diversity, and leverage that knowledge to, ideally, increase the wisdom of crowds.

The next three chapters of the thesis discuss two systems and a study, all having something in common: they leverage the diversity of domain-specific expertise represented in the crowd. They differ, however, in the type of expertise being studied. In Chapter 3, we discuss the Escalier system, which leverages fact-based criteria for measuring technical expertise. In Chapter 4, we discuss the Kurator system, which uses opinion-based criteria for measuring expertise in predicting a user's subjective

preferences. And in Chapter 5, we discuss a study of Question Finding, which uses a combination of the two: fact-based criteria to distinguish who, in the crowd, knows what, and opinion-based criteria to subjectively determine which solutions are suitable. We selected these three points on the fact-opinion spectrum to investigate more broadly the benefits of leveraging mixed expertise to crowdsourcing systems.

Having established what expertise means, the continuum of expertise that exists, and which types of expertise are of concern to this thesis, we next discuss what CSCW has taught us about designing systems that identify and use expertise.

2.2 Using Expertise in Technical Systems

To understand how mixed expertise can be used in crowdsourcing systems, we first need to understand how expertise has been used in technical systems in practice. Technical systems for expertise finding and expertise sharing in organizations have been widely studied in CSCW (see Ackerman et al., 2013, for a detailed survey). This is likely because, in practice, human expertise is a social construct[2]. By reviewing how the CSCW community has studied expertise, we can learn what things a system that needs expertise needs to consider. Organizationally, being able to easily find the requisite expertise whenever required has been studied in the domain of expertise finder systems.

Expertise finder systems (EFs), a type of CSCW or social computing system, help people locate required expertise in organizations and online communities. EF systems

---

[2] Treem and Leonardi (2016b) refer to expertise as a "communicative construct", arguing that "experts do not exist unless there is an audience out there willing to attribute expertise to them and recognize them as experts. (p. 2)" Collins (2016) formalizes this social construct in a three-dimensional model of the expertise-space, where exposure to tacit knowledge from the expert community is one of the dimensions. These are just two recent examples where expertise researchers acknowledge a sociality of expertise, but there are many more: see Treem and Leonardi (2016a) for more examples.

have evolved substantially over the last twenty years, as organizations and designers have developed a better understanding of the problems associated with the use of these technologies. What one can see is the slow but steady evolution of expertise finding in terms of both its technical and social mechanisms, as one might expect from the changes observed over time with other successful technologies (Petroski, 1994).

EFs' history shows an increasing scope of available data and the technical capabilities to handle those data in order to find others, and help the user select a suitable person for an information need. Not only have the technical capabilities increased over time, the EFs' architectures have grown to handle an increasing number of rating schemes, matching algorithms and heuristics, and selection criteria.

Below, we briefly review the progression of EF system capabilities through history, focusing on the kinds of data available and used. The discussion of available data must be inherently limited, as availability is ever-changing as more and more digital traces become available.

2.2.1 Technical Capabilities in Expertise Finding

The history of EFs shows us that EFs' technical capabilities and their ability to resolve organizational and social requirements are highly dependent on the available data. The difficulties in creating suitable profiles, especially with the additional problem of limited or reduced data (such as the email logs or hand-coded expertise maps used in the first EFs), led to the use of more and more kinds of data. The addition of new kinds of data led to better retrieval of suitable candidates. Over time, EF systems have progressively become more precise, and here we pull the key points from this progression

to carry into our work on identifying and using mixed expertise in crowdsourcing systems.

This progression of technical capabilities is evident in Merritt et al.'s (2016) review of EFs, where they describe ten representative systems that display the range of functionality and data used in EFs. Below, we briefly review what these systems brought to our understanding of how technical systems make use of expertise.

The Who Knows system (Streeter and Lochbaum, 1988) was one of earliest EFs. Who Knows used organization-specific collections of project documents, such as technical memoranda and project descriptions, to determine the expertise in suborganizations. Who Knows demonstrated that suitable organizational documents were helpful in assessing expertise at the *organizational level*.

Yenta (Foner and Crabtree, 1997; Foner, 1997) focused on finding *individuals*, introducing people with similar interests. Finding similarity of interests creates a partial EF, because a similarity of interests can imply a similarity of expertise but does not always do so. Yenta showed that documents, emails, and web posts were helpful in understanding interests and, to some degree, expertise. Its novel architecture also demonstrated the importance of considering privacy for EF systems research.

In addition to these standard types of documentation-based data, McDonald and Ackerman's (2000) Expertise Recommender (ER) also offered a flexible architecture that could support a range of expertise finding and recommendation models using different types of data. In particular, it established that using multiple heuristics for expertise might be required for an EF system to be a workable solution, in practice. ER also considered

additional social factors in expertise finding, such as candidates' suitability and availability.

The ExpertFinding Framework (Becks et al., 2004), and its successor TABUMA (Reichling et al., 2005) focused on a customizable people-matching process in an extensible architecture. Users were able to customize the matching process, which consisted of user profiles (user's education and experience) and user (learning) histories. These two systems, in addition to providing strict privacy controls, demonstrated the effectiveness of using an extensible architecture. The expertise layering technique we discussed in Chapter 1 is based on the Expertise Recommender's and ExpertFinding Framework's extensible architecture. Escalier and Kurator, which we discuss in Chapters 3 and 4, respectively, are descendants (albeit crowdsourcing-based) of these EF systems.

SmallBlue (Lin et al., 2008; Yarosh et al., 2012) represents the current state of the art for EFs. Its data included users' communication (emails and chat), content from the company intranet (blogs, wikis, and enterprise directories), as well as user-defined profiles. It then added social network-based data, "whom people know", to profiles with "what people know." Additional data included people's interests and activities compiled from the company's numerous internal sources (e.g., social networking software, company directory, geographic location, shared documents) and LinkedIn for those who had an account. SmallBlue demonstrated that social network position ("who knows whom") was important for EF, particularly within a very large organization.

The previous EFs were organizational; however, by the early 2000s, online communities were becoming increasingly important. Zhang et al. (2007) examined a way to measure expertise by focusing on the social network in online question-and-answer

communities to rank the expertise of users. The determined that a simple expertise measure was adequate, based on the assumption that question answerers are likely to have more expertise than question askers, validating the potential of EF for Q&A communities. Kao et al. (2010) and Munger and Zhao (2014) have more recently provided additional useful metrics for an EF, to include users' knowledge, reputation, authority, helpfulness, responsiveness, and sentiment.

Finally, as the societal use of social networks systems (SNS), such as Facebook or Twitter, became standard, individuals turned to them to answer questions and seek expertise. Interest was sparked in adding more formal Q&A to SNS. Bozzon et al. (2013) reported a prototype EF system, which was the first to our knowledge designed exclusively for public SNS. Their approach used Facebook, Twitter, and LinkedIn data, and they found that the inclusion of *indirectly* related resources (posts not made by the candidate expert) made the system significantly more accurate. Aardvark (Horowitz and Kamvar, 2010) further refined this by routing questions to people "nearby" in the social network. Aardvark incorporated social networks as primary data in addition to its use of expertise profiles.

In summary, EF systems through history have demonstrated the utility of using technical data at the organizational and personal level, detailed user profiles that also maintained privacy, domain-specific heuristics, social networks, social interaction data, customizable matching processes, and extensible architectures. At the end of this section, we summarize which specific lessons from the study of EFs we used for the work in this thesis. Because EF systems research has been highly dependent on the availability and evolution of data used to determine expertise, we cover that topic next.

2.2.2 Data Required to Assess Expertise

Understanding the data required to assess expertise is crucial to knowing what types of expertise measures are possible for crowdsourcing systems. Based on what we have learned from expertise finder systems in the CSCW literature, we categorized data sources into classes based on whether they directly or indirectly capture "artifactual" or "interactional" indicators of expertise. After explaining these classes of data, we walk through their advantages and limitations, concluding with a summary of what we will carry forward into our crowdsourcing work from this review.

Data used directly does not have to be converted to or reconstructed from other data before it is used in an expertise measurement. *Direct* artifactual indicators are explicit statements of expertise, and they are commonly found in self-disclosures, like someone declaring "I'm an expert in Java programming", or a credential on someone's résumé, such as a Professional Engineer license. As an example, Ackerman et al. (2003) proposed the Knowledge Mapping Instrument (KMI) to estimate people's expertise where organization members contributed questions to a company's "trivial pursuit" game, based on the knowledge required at the company. Those questions then enabled company members to estimate how others would do on the game, providing an expertise estimation of expertise under a realistic work context. Direct interactional indicators may be found in a publicized list of relational connections, such as that found in co-authored scholarly publications or networking sites like LinkedIn. Alternatively, Farrell et al. (2007) used social tagging (e.g., categorizing people based on the their projects) and found it effective for characterizing people's expertise.

Indirect measures attempt to characterize the same information represented by direct observation, except this information typically must be inferred from a person's activity. In the EF literature, a person's activity has generally been measured through two behaviors: what people create (artifactual) and with whom people communicate (interactional). Algorithms are used to turn documents or other artifacts (e.g., code, forum posts, or email content) into a mapping of people to keywords and topics. These artifactual indirect metrics measure what someone knows. Algorithms can also characterize a person's social network based on communications with others (e.g., email or question-answer forums); these become indirect interactional metrics when they are used to determine an entire network of relations (question-answering, communication, work projects, and so on).

It is important to note that the artifactual "what someone knows" and the interactional "whom someone knows" are analytical distinctions, since the two are often conjoint. In addition, some systems use mixed metrics for indicators of expertise. For example, Munger and Zhao (2014) applied sentiment analysis to Q&A posts to characterize whether an answer had a positive or negative sentiment (positive sentiment favorably influenced the answerer's expertise score). Availability in SmallBlue (Yarosh et al., 2012) and responsiveness in Kao et al. (2010) and Munger and Zhao (2014) of candidate experts are more examples of features used to measure helpfulness. Indeed, in Kao et al. (2010), a person's "reputation" metric includes availability, responsiveness, and sentiment-based features.

Some initial EF systems used direct artifactual indicators. However, people's self-reports are unreliable (Donaldson and Grant-Vallone, 2002 and Arnold and Feldman,

1981). There are strong organizational, as well as personal, reasons to either hide or over-promote expertise. There are relatively few observational measures in organizations except for managers' periodic assessments, and these, too, can be unreliable.

Indirect artifactual indicators solved aspects of these problems; indirect artifactual indicators could construct expertise profiles without intervention. Expertise profiles could be constructed from project reports or later, Intranet documents. The list of potential sources has become larger and larger over time.

Indirect measures solve two additional data problems. The first is the standard problem with explicit data collection: users' lacking sufficient motivation to enter their data (Hinds and Pfeffer, 2003). The second is related: it is the problem of maintaining data over time, which requires the continued motivation to enter one's data accurately (Ehrlich, 2003). This can lead to non-consistent entry of data, incomplete entry of data, and out-of-date data. While incomplete entry and perhaps non-consistent entry can still result in useful EF systems, out-of-date data quickly lead to disuse. If an EF cannot be used for current needs or points to the wrong people, it is less than helpful.

Indirect measures, while they solve those problems, have other problems. While standard information retrieval relies on incomplete or ambiguous representations of information sources, EF metrics use even more incomplete or ambiguous representations. While indirect measures solve self-report errors, they introduce many other sources of error. The issue of inferring topics from keyword vectors or even more suitable representations (e.g., topic models or LDA) is well understood in the information retrieval community, and there are the difficulties of distinguishing mere interest in a topic versus expertise in that topic and of determining relative expertise. These issues can

introduce error into an EF. Systems generally ameliorate this concern by producing a candidate set which can then be manually examined.

Indirect measures also suffer from the problem of new employees and their cold start. New employees obviously will not have built a repertoire of suitable content. This issue introduces a form of incompleteness into the EF system by reducing the potential candidate set. It is not easily corrected, although in an organization of a sufficient size, this issue is likely to have negligible effect.

More importantly, indirect measures still rely on incomplete data. Not everyone has published material in the public domain or on organizational Intranets. Not everyone even produces documents, even though they may have a great deal of organizational expertise (e.g., admins). Invisible work (Star, 1999) is, indeed, often invisible and not captured by EF systems.

There has been recent work on discount expertise metrics that attempts to sidestep the problems we see with indirect measures of expertise. Hung and Ackerman (2015) found web browsing history to be useful in reliably determining if a person were likely to be an expert or novice in programming. Even though their approach yields a coarse-grained stratification of expertise, the ubiquity of web browsing makes this a promising step towards alleviating some of the challenges in using indirect measures. In Chapter 5, we discuss a third project called Question Finding, which includes an attempt to more generally address the problem of discount expertise metrics.

2.2.3 Summary

Technical systems that find and use expertise has been well-studied in the CSCW literature, particularly in the domain of expertise finder systems. In designing the mixed

expertise crowdsourcing systems presented in this thesis, we used some of the lessons learned from our review of expertise finder systems. Specifically, we found that extensible architectures are effective in practice because they allow for customizability and flexibility depending on available data, heuristics, and expertise identification and matching algorithms. In short, extensible architectures are an imperative for environments with constantly evolving data availability and algorithms, which is true of crowdsourcing environments. Using this lesson, Escalier and Kurator both use extensible architectures. Escalier allows for arbitrary discount expertise metrics to be used in its expertise layer, but the system uses a fixed number of layers (i.e. a base layer and an expertise layer). Kurator takes this idea further, allowing for an arbitrary number of layers. It uses family, crowd, and machine learning layers by default, but the system can handle multiple machine learning, crowd, and family layers.

As well, we learned that technical systems using expertise are limited by the data that is available to measure that expertise. Direct indicators of expertise are often unreliable and impractical to maintain. Indirect indicators alleviate the need for direct reporting from users, but it introduces the additional problems of being inconsistent and incomplete. Discount expertise metrics is a promising direction, but researchers have only very recently begun to focus on this. After reviewing the relevant crowdsourcing literature, which we do next, we return to this discussion of what data is available to crowdsourcing systems.

2.3 Using Expertise in Crowdsourcing

This thesis is focused on the study of crowdsourcing *systems*, which we define as using crowds within an automated workflow. This is not quite the same as human

computation, depending on how the reader interprets Quinn and Bederson's (2011) taxonomy, because we do not want to constrain our explorations only to systems where "human participation is *directed* by the computational system (p. 1404)" (emphasis mine). Escalier, the first system in this thesis and discussed in the next chapter, allows a community of people to explore a space as they see fit, and the backend automation makes inferences based on human behavior and presents new information back to the users. This does not fit cleanly into a human computation paradigm, but it does fit into our definition of crowdsourcing.

As mentioned, many crowdsourcing system designers take a binary perspective on expertise and assume the crowd consists of non-experts in its problem domain, ostensibly because designers believe the aggregate wisdom of crowds (Surowiecki 2005) can approach that of an expert. This approach is pragmatic and is proven to be effective for many problem domains. As a relatively early example, in Snow et al.'s (2008) work with natural language processing, they use non-expert crowd workers to provide judgments on multiple tasks, such as finding similar words and assessing emotional valence of news headlines, and the crowd's responses showed high agreement with expert labelers. There are many other examples of systems designed to use non-expert inputs to replicate expert-level work in the domains of:

- *Computer vision*; Sorokin and Forsyth (2008) acquired annotations for a large number of images by utilizing a crowd's latent ability to recognize simple images. Annotated image datasets were, and still are, important to computer vision research, and prior to leveraging the crowd for annotation, researchers typically acted as the "expert" annotators for image datasets.

- *Document editing*; Bernstein et al. (2010a) leveraged a non-expert crowd for their "basic knowledge of written English" to help novice and expert writers offload word processing tasks such as shortening paragraphs and formatting, spelling, and grammar checks. Their find-fix-verify workflow garnered enough quality from the crowd to alleviate the need for expert human editors.
- *Visual question answering*; Bigham et al. (2010) introduced a crowd-powered system that answered open-ended, natural language requests from blind users. The crowd's basic abilities to see and write were effective enough to outperform, in terms of monetary cost, many automated tools built specifically for this purpose.
- *Translating text*; Hu et al. (2011) designed a crowdsourcing system to support monolingual translation, which uses people who know only the target or source language of a machine-translated text. This allowed many crowd workers, who did not have bilingual (or multilingual) expertise, to contribute in the domain of text translation.
- *Real-time systems*; Bigham et al. (2010) established the idea of near-real time responses from the crowd, and Bernstein et al. (2011b) extended the idea to the concept of synchronous crowds for on-demand crowdsourcing (<2 second response time). Their approach used the crowd's collective ability to identify the "best" moment from a short video. Similarly, Lasecki et al. (2012) leveraged real-time crowdsourcing to allow deaf people to request speech captions on-demand. In their own words, their system "enables non-experts to contribute without any special training or skill. (p. 3)"

- *Creative work*; Lasecki et al. (2015) created a crowdsourcing system for Wizard-of-Oz prototyping of user interfaces. Their system allowed crowd workers to collaboratively sketch and iteratively improve sketches of a shared user interface. This approach applied the idea of real-time crowdsourcing systems to highly creative work to alleviate the need for a single expert user typically needed for Wizard-of-Oz prototyping.

These systems are examples of common domains of research within crowdsourcing that all have something in common: their reliance on non-expert crowd abilities to perform work typically done by experts. Leveraging *mixed* expertise in crowd work has been under-studied in the prior literature, and it is rarely explicitly framed as such.

There have been, to our knowledge, no systematic examinations of how to leverage different types and levels of expertise in crowds. However, there has been work in crowdsourcing systems where experts or expertise is a focal point. In these systems, the social arrangement of the workers and requesters is important and has influenced the architecture of the systems. Thus the first part of our review of the crowdsourcing literature is organized according to the social arrangements at play. The second, and final, part of the review discusses crowdsourcing systems in the context of the types of expertise they use.

2.3.1 Social Arrangements of Expertise in Crowdsourcing

The expertise-focused crowdsourcing literature can be divided into three categories of social arrangements, each representing different ways to account for input

from expert and/or non-expert workers: using input only from expert crowds, having experts guide non-expert crowds, and mixed arrangements.

2.3.1.1 Using Only Expert Crowds

First, there are systems designed to focus solely on expert workers. Chilton et al. (2014) introduce Frenzy, a conference planning tool designed for large groups of experts to collaboratively build a conference program. Instead of tasks being routed to experts, experts self-select tasks (papers) based on their topics of expertise. Frenzy provides a way to facilitate experts working simultaneously on a complex task.

Similarly, Foundry (Retelny et al., 2014) enables expert flash teams, where multiple experts come together to quickly and collaboratively work on modular tasks that can be linked to other modular tasks. Multiple tasks can be combined to accomplish highly complex tasks, such as video animations, mobile web applications, and platforms for online educational courses. The work is performed by experts regardless of who requests the work.

Kulkarni et al.'s Wish system (2014) allows expertise to be solicited when a non-expert user lacks what he needs for specialized, creative work. The Wish system uses MobileWorks, a commercial platform, to post wishes and solicit experts for help. The expert selection process is also handled by MobileWorks, where an expert identification algorithm is executed to recruit new experts only when the initial wish goes unanswered.

The work in this thesis adds to this prior literature by not focusing solely on expert crowds. Instead, we hope to identify various levels of expertise within crowds and to leverage the full spectrum of expertise where possible.

2.3.1.2 Experts Guiding Non-expert Crowds

Second, instead of relying solely on expert crowds, there are workflows designed to allow experts to guide non-expert crowds. In an early study of this, Kittur and Kraut (2008) examined Wikipedia's contributions and coordination amongst editors and found that the timing and type of coordination used affected the quality of an article. In particular, they conclude that concentrating the complex tasks, such as establishing structure and cohesion during article formation, into fewer editors would likely lead to higher article quality. Although it is not stated explicitly in the paper, it is implied that these few editors possess a high level of expertise on the article's topic. This is an early example of experts guiding a crowd of contributors by laying down the structure of a Wikipedia article during its formative stage.

Dow et al. (2012) use a shepherding metaphor to demonstrate how task-specific, external feedback, provided at the right time, increases the quality of work provided from the crowd. Chan et al. (2016) expand this shepherding concept into idea-generation, creative work. They show that experienced facilitators are able to help workers come up with more ideas, and these ideas were more creative than the ideas of workers who were unfacilitated. Kim et al. (2014) use a similar metaphor of leaders directing followers, and they apply it to collaborative story-writing. Leaders served as the expert facilitators of the stories by directing the work and communicating goals to collaborators.

The systems presented in this thesis build upon this prior work by using a combination of experts guiding the work of the crowd while simultaneously performing tasks alongside the crowd.

2.3.1.3 Mixed Arrangements of Expertise

Finally, there are few crowdsourcing systems acknowledging an approach designed specifically for a mix of expertise in worker contributions. Law et al.'s (2013) Curio platform is designed to help scientists, who may be non-expert in implementing crowdsourcing projects, to use crowdsourcing. It is a system that allows expert requesters (i.e., scientists) to direct and control crowdsourcing projects that use amateur workers (i.e., citizen scientists). Curio allows for hierarchical teams, where researchers can select participants with specific expertise as well as collaborators with significant domain knowledge to be on the project management team. This work explicitly calls out "mixed-expertise crowdsourcing" as the combination of expert and amateur workers or requesters.

In a different take on mixed arrangements of expertise, Huang et al. (2015) designed Guardian, a crowd-powered spoken dialog system, to use inputs from a non-expert crowd to filter out "unnatural" parameters from various web API's to lower the threshold for programmers (i.e., the experts) to contribute to Guardian. With this arrangement, the non-expert workers are leveraged earlier in the workflow, laying the groundwork for expert workers to contribute more easily on a separate task.

Curio and Guardian both use social arrangements that allow for some non-expert participation, but they emphasize expert (or near-expert, in Curio's case) participation because the problems require it. Those experts are known to exist and are accessible. In this thesis, we are interested in problem areas that are not as mature as these, where expert crowds are unidentified or inaccessible and cannot be relied upon, at least at the outset of the system deployment. Thus the reliance on less-than-expert input becomes

paramount, which makes these systems unsuitable for our purposes. The work in this thesis adds to the dearth of crowdsourcing literature on mixed expertise by using a greater diversity and depth of mixed expertise.

2.3.2 Types of Expertise Used in Crowdsourcing

Most of the crowdsourcing systems reviewed up to this point either pre-determine expertise by the social arrangement used, or it focuses on fact-based tasks (as mentioned, computer vision, document editing, etc.). However, few focus on opinion-based tasks. Some researchers doubt that certain technologies, crowdsourcing in particular, are applicable to problems having personal or highly subjective aspects to them. This sentiment is summed up by Simko and Bieliková (2011, 45): "Automated or crowdsourcing approaches are inapplicable in [the] case of personal content or content of a small social group (e.g. family)." The pervasiveness of this sentiment is unclear, but Organisciak et al. (2014) acknowledge that researchers in crowdsourcing have only recently begun *focusing* more on problems with a "subjective aspect to them." Although there has been crowdsourcing research in subjective problems (e.g., word processing in Bernstein et al., 2010a, itinerary planning in Zhang et al., 2012, and managing email in Kokkalis et al., 2013), Organisciak et al. (2014, 193) formally call out this "class of problems where the task is time-consuming for an individual, but its subjective nature makes it difficult to delegate."

The Kurator system, discussed in Chapter 4 of this thesis, is used to study the utility of crowdsourcing for a highly subjective task: personal digital media curation. One goal of Kurator is to extend the crowdsourcing literature to new kinds of subjective

problems and, in doing so, to demonstrate that leveraging the diversity of expertise in the crowd is an effective way to begin to solve subjective problems of this kind.

2.4 Summary

In the previous sections, we discussed the importance of considering a continuum of expertise in the crowd, and that the crowdsourcing literature does not address this continuum. We have also discussed that the fact-opinion spectrum of criteria for measuring expertise is a useful framing for exploring this space in crowdsourcing research. As well, we touched on the importance of using extensible architectures when building mixed expertise systems. Finally, as mentioned, we return to the discussion of the data available to crowdsourcing systems.

The data used to measure expertise is plentiful in CSCW and almost non-existent in crowdsourcing. Organizational and Internet-scale expertise finder systems, as discussed, use direct and indirect artifactual and interactional data. Crowd work has limited availability of these data due to the nature of anonymous microtasking environments as well as a lack of focus on obtaining or using such data. There is little interactional data to use because this would require access to workers' interactions, which is almost non-existent in microtasking environments. Thus the focus is on artifactual data--what workers produce and how they perform. When expertise is determined dynamically (which is rare, as discussed above) direct artifactual indicators are relied upon almost entirely. In microtasking environments, a worker's success rate is a direct measure of their ability to perform tasks well, in general. For a richer, albeit still flawed, representation of expertise, qualifications or credentials can be awarded based on

successful performance on specific types of tasks (see Mechanical Turk). These are all direct artifactual indicators of expertise.

Expertise finder systems also use the social interactions between knowledgeable actors as interactional indicators of expertise. By design, in microtasking environments, it is difficult for crowd workers to engage in such social interactions. The crowdsourcing and human computation communities are only now looking at ways expertise can be shared on real-time microtasks (e.g., Lasecki et al., 2015). In general, crowdsourcing systems lack support for the social needs of workers, which may make the systems less efficient. Implementing social interactivity, and leveraging the metadata of that activity for expertise measurement, is a potential future direction for the line of research presented in this thesis. However, as social interactivity is incorporated into crowdsourcing systems, privacy issues are surfaced, as we have seen in the review of expertise finder systems.

Finally, exploring the use of discount expertise metrics is a promising direction for both the CSCW and crowdsourcing literature. Escalier is able to use discount expertise metrics from Hung and Ackerman (2015) and Zhang et al.'s z-score (2007). Our study on Question Finding is a deeper investigation into using discount expertise metric for specific topics, such as space science.

In this chapter, the crowdsourcing systems we reviewed, in almost all cases, used predetermined experts, or experts who were handpicked for experiments. Where expert crowds are known and accessible, this is a non-issue. However, as we show in this thesis, considering the mix of expertise within a crowd can significantly improve the effectiveness of certain crowdsourcing systems, particularly in new problem areas. In the

next chapter, we introduce the Escalier system, which builds on the lessons learned in this chapter's literature review, and it answers the first research question in this thesis.

Chapter 3. Escalier: Mixed Expertise within a Crowd

In Chapter 1, we described the importance of considering the diversity of expertise present in the crowd. We refer to this diversity as mixed expertise. Our aim, ultimately, is to be able to differentiate this mix of expertise in order to leverage it to improve system performance.

This chapter lays the foundation for designing systems with mixed expertise in mind. In this chapter, we introduce Escalier, a crowdsourcing system based in social navigation that uses mixed expertise, and through a simulation-based evaluation, we quantify the material benefit of using a range of expertise levels from the same crowd. In this work, we assume expertise levels can be identified because we first need to establish if *using* mixed expertise is worthwhile. Escalier is a solution within the first problem area discussed in Chapter 1, where the diverse expertise of a population can be leveraged to uncover more of a solution space. Through our study of Escalier, we answer the first research question:

- **RQ1**. Under what conditions, and to what extent, does mixed expertise within a crowd materially benefit a crowdsourcing system at scale?

Although not explicitly called out in the narrative that follows, we used the expertise layering paradigm discussed in Chapter 1 in constructing Escalier. We obeyed the two principles of expertise layering: 1) adding expertise, through expertise

assessments, should add value to the system when available, but the system should not be *required* to use expertise (i.e., the use of expertise is non-blocking), and 2) expertise use should be *modular*, where different mechanisms for assessing and using expertise can be added or swapped out. The current implementation of Escalier uses fact-based expertise assessments, which we discuss in more detail below.

3.1 Introduction and Background

Social navigation is something we use often in the physical world (Dieberger et al, 2000). Social navigation occurs when we use the activity of other people as cues for our own decision-making, such as hiking on an uncharted path but following a trail, or using the size of a crowd outside a restaurant to help you decide what the good restaurants are. In the context of the work presented in this chapter, social navigation is a form of *passive crowdsourcing* (Bigham et al., 2015), where crowd activity is not *directed* but rather monitored and leveraged. Social navigation occurs when someone uses the activity traces left by other people to inform their decisions. A social navigation system makes a crowd's activity traces visible to an end user, or it makes inferences on the traces directly. One such popular system is Aardvark (Horowitz and Kamvar 2010), which used social navigation to help it route questions to answerers in a user's social network.

The idea of social navigation in the digital domain has been around for decades. A seminal work in this domain is Hill et al.'s (1992) idea of "computational wear", where the history of editors' or readers' interactions with a digital document is shown graphically in the document's scroll bar. Wexelblat and Maes (1999) build on this work by applying the same principles to web navigation. Freyne et al. (2007) reprise the work

on web navigation by combining the ideas of social search and social navigation to grow and use "community wisdom" in web searches. Dieberger et al. (2000) was one of the first to define social navigation traces as a *passively* grown by-product of how people use a space. Digioia and Dourish (2005) expand on this notion of passivity in activity traces by arguing for users to have freedom to interpret traces however they see fit, which they apply to usable security. Goecks et al. (2009) extended this work to help users make decisions about their privacy and security configurations.

It is our hope in this chapter to understand how user expertise, revealed as fact-based measures in system use and as explicit ratings, can be better incorporated into crowdsourcing systems, in general, and social navigation systems, specifically. Accordingly, this chapter makes two contributions to the crowdsourcing and social computing literature:

First, we demonstrate through a system design, and then show through a study, that using people's expertise levels as part of crowdsourcing can have significant benefits for at least some applications.

To do this, we present a system we have built. Our *second* contribution is Escalier[3], a crowdsourcing system that shows there are possibilities of using a diversity of expertise explicitly in crowdsourcing. It does this by actively aggregating activity traces. Escalier was explicitly designed to demonstrate that accounting for mixed expertise in the crowd could help the effectiveness of a social navigation system. We believe Escalier is

[3] My contribution to Escalier, the system, is in creating version 2; version 1 was tested heavily, and the applications discussed in this section were created with version 1; version 2's re-write includes, among other things, the scripting layer of Escalier.

novel in its own right and may lead potentially to a range of other crowdsourcing systems.

Below, we first present an example use scenario, then the formalisms underpinning Escalier, and then the Escalier system itself. We follow that with a brief overview of simulations as an evaluation technique. We then present the results of a 4x4x5x2 set of simulations, plus additional sensitivity analyses, that examine the basic workability of Escalier and the utility of using expertise. We follow with limitations and future work, and then conclude.

3.2 Use Case

It is easiest to understand the Escalier functionality from the application side. An example of how Escalier could be used is the Am-I-Normal (AIN) application, which helps users with their system configurations. AIN highlights an important problem for end-users: they need technological support for their individualized use of technology (Huh et al, 2011). Specifically, AIN addresses the issue that people may or may not understand whether they have unusual system components, especially software drivers. Many users do not update their drivers or other system components reliably. Even those who update consistently may choose to wait until new drivers have been patched and are reliable. Knowing when to update can be difficult.

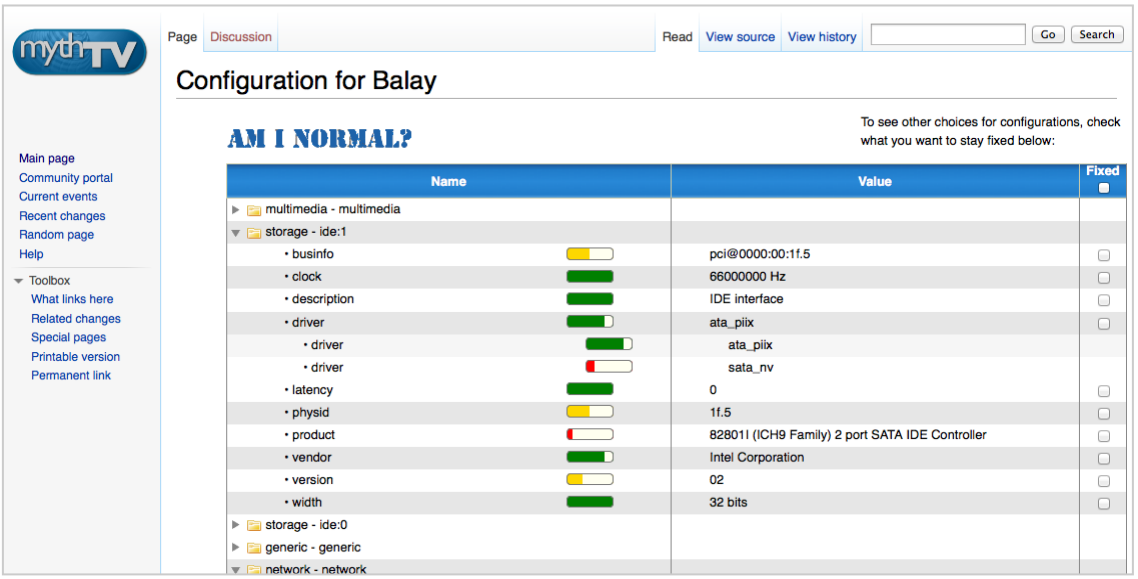


Figure 1. The Am-I-Normal (AIN) application. AIN lets users know whether they have unusual configurations. AIN uses Escalier to obtain "conformity" data to display to the user.

In Figure 1, Balay is using Am-I-Normal, a web app, within his community's MediaWiki, in this case, for MythTV, a Linux media platform. AIN tells a user how common each part of his system configuration (software components, drivers, hardware devices, etc.) is for all users. (For the AIN prototype, the data came from a dataset of 330 Linux computers.) Commonly used components are shown in green. Note that for Balay one of the "storage - ide:1" drivers, called "sata_nv", is unusual. It is shown in red, indicating that a small percentage of users have this driver. For Balay, it is probably time to update the driver or at least to investigate why it is uncommon. On the other hand, the user can see that the rest of his drivers are used by many others and are likely to be reliable if not up-to-date. True to the intent of social navigation, the aggregated information from a crowd's activity is made visible to the user in a meaningful way, and AIN itself is not directing the user towards any particular decision.

AIN is made possible by Escalier's knowing the system configurations of many other users. For AIN, Escalier merely needs to know that the system configurations exist. More complex applications require a greater understanding of how configurations meet an objective criterion, and Escalier can also provide this. Next, we describe the formalisms that form the foundation for how Escalier is built.

3.3 Escalier Formalisms

Escalier creates a crowdsourced map of what configurations "work" – in the view of its users. Escalier is crowdsourced in the sense that its many users co-construct its map by exploring single points or even subspaces.

Users report on whether their configurations "work" based on some objective function. Of course, users' reports may vary in reporting whether something "works" or meets its objective function. Therefore, Escalier's co-constructed map is probabilistic because any given user report could be incorrect, so Escalier has only a likelihood estimate that the given configuration of settings or traces meets the criterion. Below, we describe how a likelihood estimate is made. Before explaining the architecture and components of Escalier, we first define Escalier's features formally. We will necessarily stay very abstract at first, but we will follow the abstractions with an example application.

We will term a set of settings or activity traces as a *configuration*, denoted as c. (Note we are distinguishing between "configuration," which is a generalized abstraction in Escalier, from "system configuration," which is used in computer systems management.) Configurations are defined as ordered lists of properties, p. The set of all properties in configuration c is $P_c = \cup\{p_i\}$, where p_i is the i'th property in configuration c.

In turn, each property has some value v, and therefore v_i is the value assigned to the i'th property p_i. The set of possible values for a property is $V_i = \cup\{v_i\}$. Note any given V_i could be very large or even unbounded.

Formally, then, a given configuration c is defined as the ordered set of existent (property name, property value) duples, or $\cup(p_i, v_i)$ for $\forall_i$ in c. C is the set of all possible configurations.

Escalier's goal is to determine which c in C meet some criterion, such as exceed a threshold value for an objective function. We will term some c that meets an application criterion as a valid configuration. In domain-specific uses, the general term *valid* is likely to be replaced with a domain-specific term. For example, security applications might also use the term secure settings, and a valid configuration in an activity trace-based domain might be a relevant trace. Domains concerned with system configurations would typically use the terms stable and stable system configuration.

To find valid configurations, Escalier assumes a constant stream of user reports. A user report consists of submitting the configuration c, and expressing whether it meets an objective function. This user report may be explicit or inferred. For example, the user can explicitly report whether she believes her system is stable or not. The user report can be automatic, such as when the system itself reports whether it has crashed with certain settings. However, under many conditions, the outcome may need to be inferred. For example, in the case of a system configuration without crash data, a report may consist of whether a system configuration has been selected and kept for a period of time above a threshold. For activity traces, as suggested by (Lee et al. 2013), a configuration will be a trace that has been either labeled or assumed to be relevant.

In summary, Escalier lets users know what configurations in the space C fit their goals by each person reporting their successes and failures, collaboratively constructing a map. It functions as an online algorithm, generating its map as a constant stream of user reports comes in. Therefore, Escalier serves as a collaborative reinforcement learning platform (Barto 1998). Unlike reinforcement learning, however, each user explores a point or subspace in the space C, the space of all possible configurations. As more users report positively on a given c, the more the likelihood of that c meeting the given criterion is reinforced. Because reinforcement learning and social navigation (Wexelblat and Maes 1999) are similar mechanisms in their abstractions, Escalier also enables a new form of social navigation, one that is crowdsourced and probabilistic.

Next, we explain in detail how Escalier works, how the likelihoods are estimated, and how users' levels of expertise can be leveraged.

3.4 Escalier Platform

We now provide an overview of the system itself and how it works. This section will also detail how the likelihood estimates are made and used.

To make our explanation clearer, we will continue our use case of system configuration to explain how Escalier works. It is a simplification of actual system configuration; see (Chau et al. 2011) for a system designed to handle system configuration specifically. Instead, here we walk through the use case only to show the functionality that Escalier can provide, assuming that there is additional support for any given domain that may be required.

Below, the Escalier core functionality and its support for prototype applications will be detailed in turn.

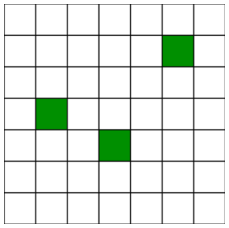
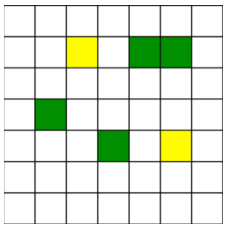
(a)  (b) 

Figure 2. How the Base Layer functions. (a) Escalier starts up with canonical (test) cases that are known to be valid (green). (b) Users report over time, growing the number of valids (green) and nearly-valids (yellow).

3.4.1 Escalier Core

In general, Escalier consists of a standard web-based service architecture connected to a database, a set of applications, and auxiliary maintenance services. This architecture allows separate applications to provide customizable end-user functionality.

Escalier conceptually consists of two layers, where the functionality of the second layer builds on the first layer. The layers are:

3.4.1.1 Base Layer – a Bayesian map.

The first part of Escalier keeps a probability assessment for the likelihood of whether every potential configuration meets its objective function and is therefore valid. For system configuration, it would be whether that system configuration was stable; i.e., it does not crash, or it appears to the user to be stable.

The map is initialized with what we call canonical configurations. These are configurations known to be valid. They may be, for example, system configurations that a vendor or open-source community heavily tested as part of the release cycle. Or, they could also be hand-labeled activity streams or security settings.

Thereby, Escalier starts with some known configurations that meet the objective function criterion. These canonical, known valid configurations are shown in Figure 2(a)

as green cells in a notional 2-dimensional space of all C. Starting with known solutions avoids the standard cold start problem (Adomavicius and Tuzhilin 2005), i.e., the problem with many social computing systems of not having data when they initialize. It should be noted Escalier can start with a small number of canonical configurations initially and still be effective; we will provide test results below.

As more and more configurations are reported to Escalier, these reports update the probability assessments for specific configurations. If the user report indicates the configuration meets the objective function (either by explicitly labeling the configuration or more likely it being automatically inferred), the likelihood is increased. Conversely, if the user report indicates the configuration is invalid, the likelihood can be decremented. Over time, this constructs a map of the configurations inferred to be valid.

A configuration is given an a priori value (hence it is a Bayesian map). As users report, the probability assessment for any given c_i is increased or decreased by a value. If the probability assessment for c_i eventually exceeds a pre-defined threshold, then Escalier will report c_i as meeting its objective function. One way to do this is to have all users' reports be worth the same value. Another way is to weight values based on the expertise of the user and/or her history of reports. Weighting of reports based on expertise is handled in the Expertise Layer, discussed below. We believe that report values used by Escalier for its likelihood assessments may need to be tuned by domain and community, but we have not tested this.

Figure 2(b) shows the Base Layer after user reports have come in. Remember, we do not know whether the user reports are accurate, and so validity is conditional on other users' reports. If enough users have reported positively about a configuration so that its

validity assessment is above a threshold, then Escalier believes the configuration meets the objective function. This is shown as a new green cell in Figure 2(b). If a user, for example, wanted to know what configurations were valid, Escalier would report that any configuration above the threshold (green) was valid and any configuration below it (yellow or white) was invalid.

To continue the systems configuration example, over time, users report on the stability of their system configurations. An Escalier application, described below, can provide this relatively simple functionality for uploading configurations. These user reports are time-stamped, and our Escalier prototype assumes that if users stay with a configuration, it is likely to be stable. As more users land on a system configuration, then there is a higher likelihood that the configuration is stable. If users leave a configuration, it may be unstable or it may not have provided the desired functionality, and so the assessment is decremented lower. Thus, the crowd's use ultimately leads to accurate assessments about the validity of system configurations.

3.4.1.2 Expertise Layer – a Bayesian network of expertise assessment.

In its default state, the Base Layer is a straightforward probability map, where every user's report is weighed equally. However, there are often users with more expertise. The Expertise Layer, the second part of Escalier, weighs the likelihood assessments by the expertise of the user that made the report. The Expertise Layer consists of another map that assesses users' expertise levels, and its values are used to weigh the increments (and any decrements) in the Base Layer.

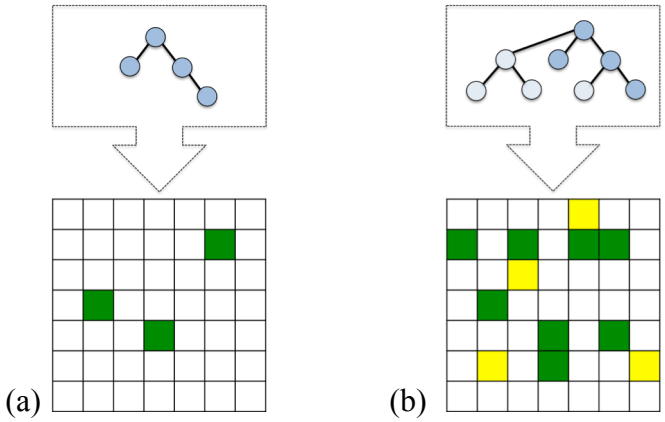


Figure 3. How Expertise Layer functions. (a) Escalier starts up with knowledge of user expertise from Q&A community data, for example, in the Expertise Layer (top). (b) Users report over time, and these reports are weighed by the expertise assessments (additional expertise data may also be added over time).

The Expertise Layer is dependent on obtaining users' expertise levels. This can be done in a number of ways, none of them perfect. We have constructed two different Expertise Layer mechanisms. The first uses a simple z-score metric based on the metric in Zhang et al. (2007). This z-score is calculated from who answers whom in a Q&A community, allowing one to infer a 5-level expertise rating. The ratings are dynamically adjusted as Q&A activity continues, so the z-score-based Expertise Layer is constructed as a Bayesian network. Using z-scores, however, requires Q&A community data. Since most users will not participate in some Q&A community, this approach requires that we label these users' expertise rankings as novices or don't-knows.

The other expertise metric we constructed is based on a user's web history. Hung and Ackerman (2015) found that one's web history of searching websites can identify users who are novices and who have high expertise in technical subjects. We believe that simple metrics, such as the use of advanced commands in Linux, can also imply a high-level of expertise. (We note that all of these can be calculated on a user's machine and are therefore privacy sensitive). We are confident that suitable discount expertise metrics can

be found, although we believe different domain areas, such as security, will have other, specialized metrics.

The expertise evaluations lead to a weighting factor for users' reports, and the Expertise Layer provides reweighting for the Base Layer's likelihood assessments, as shown in Figure 3. The intuition here is that reports from someone with the highest expertise should be trusted, those from novices substantially less so. Without the Expertise Layer, the Base Layer will rate each user as a novice. As we will see later, adding expertise information substantially improves Escalier's performance.

Weighting by expertise also provides additional protection against poor false positive assessments. Escalier's Base Layer can be tuned to be conservative, as it is in our current prototype. However, because it is conservative, it requires many more reports than would be needed if one could weed out erroneous reports. The Expertise Layer provides the ability to winnow the end-users and pay more attention to those with higher levels of expertise or experience.

In terms of our example use case, the Expertise Layer would have assessments of people's technical expertise. We could assume those with higher expertise would be more likely to know when their system configurations were stable and when to look for new settings.

Note that if more data were available (e.g., crash reports), Escalier's assessments become only better. As well, we remind the reader that Escalier can say something about stable configurations from the very beginning because of its canonical data (Base Layer).

3.4.2 Escalier Applications

Escalier consists not only of its core functionality, but also support for applications. We have designed a number of Escalier applications. One was presented above; it was the Am-I-Normal application that determines whether a user's system settings are consistent with the crowd's. We have also constructed a simple reporting prototype. This application can also reside within a community MediaWiki. It is designed to give an incentive to report one's configuration. The reporting application uploads a user's configuration using a client-side script. It omits or obscures any privacy- or security-sensitive data. It also reports back to the user whether her configuration is likely to be valid. The reporting application can run stand-alone or, as with the MythTV forum, in a community wiki. Having applications run in the community's wiki will make it easier for users.

Another application is Whats-Next (Figure 4), again incorporated as part of a MediaWiki page for a community in a simple recommender that allows a user pivots on her own configuration or a subset of it. Using What's-Next, the user determine what additional system components could be added, with Escalier saying which new system configurations are likely to be stable.

3.5 Implementation

Escalier consists of both Java code, which is the real kernel of Escalier, and Python, which provides scripting that adds functionality to the Java layer, supports prototyping, and handles the server-side of applications. The functionality of the MySQL engine is used extensively, so Escalier itself is relatively small. Currently there are about 4500 lines of Java and Python in the Escalier system. The 3 applications presented in this

paper, relying on functionality on the Escalier side, are only approximately 500 lines each. Additional applications can be easily supported by adding application-specific functionality to Escalier's scripting.

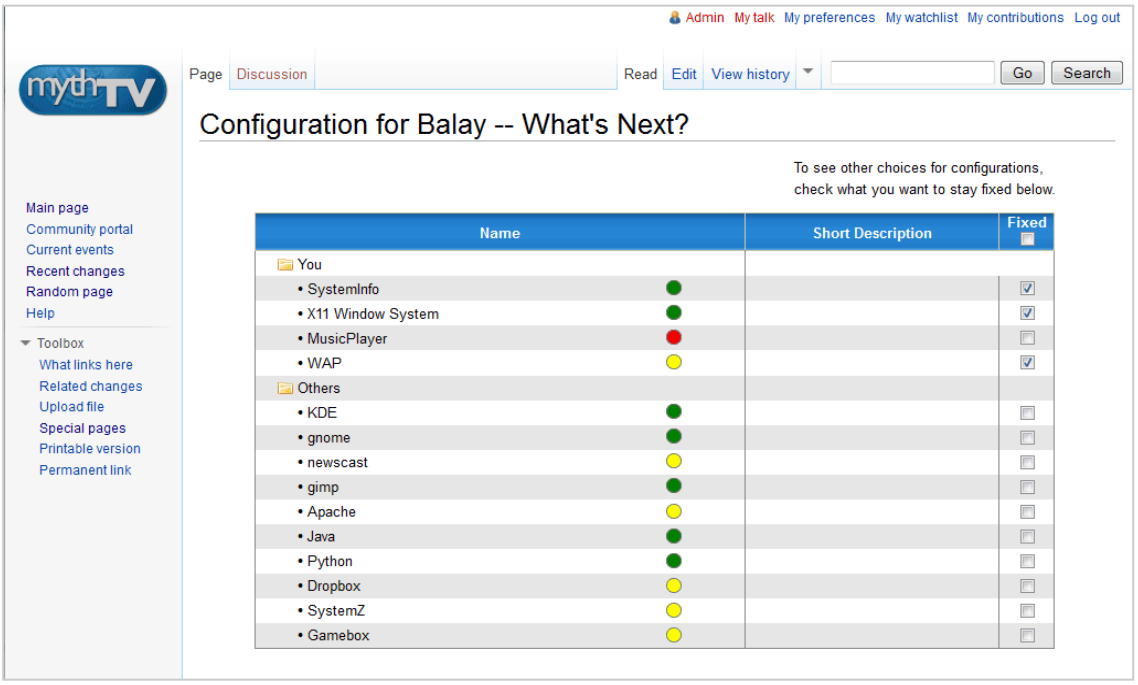


Figure 4. The What's-Next (WN) application. WN lets users look at what components they can add or modify, but still remain stable.

3.6 Evaluation Through Simulations

We chose to examine Escalier's potential workability, especially in terms of its use of expertise, using simulations. While preliminary tests of the Am-I-Normal and Whats-Next applications indicated Escalier's potential, to assess Escalier's actual performance at scale was intractable. We would have had to not only find tens- to hundreds-of-thousands of users, but also do so under conditions where we would not yet know whether the system was useful. This is not an uncommon problem in crowdsourcing research, and it inhibits finding new crowdsourcing solutions.

Instead, in the remainder of this section, we explore the use of simulations. We wanted to know:

- Whether Escalier is feasible in principle and under what conditions
- Whether the use of expertise leads to better crowdsourcing results.

To answer these questions, we used simulations to examine the system at scale. The next section explains an approach for doing so.

3.7 Why Simulation

In simulation, one needs to construct models to examine the phenomenon in question. Each model, as (Gilbert and Troitzsch 2005) states, "...is a simplification – smaller, less detailed, less complex, or all of these together – of some other structure or system. (p. 2)". One does this to better examine the relevant effects of important system and user characteristics and, in our case, variables such as number of users, density of the configuration space, user expertise, and so forth.

While there are many types of simulations, we decided to use a hybrid between simulations of computational systems and agent-based modeling and simulation (ABMS). (It is likely that simulations of crowdsourcing systems will be a hybrid.) The former has a long history in computer science (MacDougall 1970), although they are seldom used in HCI and CSCW. These simulations have primarily modeled the throughput of hardware and then the performance of software systems, including networking and distributed systems. Entities being modeled might include, for example, mainframe jobs that require disk and memory allocations. ABMS, on the other hand, has been used extensively in the social sciences as well as in distributed artificial intelligence (Bandini, Manzoni, and Vizzari 2004). In ABMS, agents consist of models of intelligent and social behavior; these agents operate in some modeled environment.

Our simulation of Escalier as a crowdsourcing system must share many characteristics of software system simulations. Our goal is to understand the characteristics under which Escalier functions best (to be defined below). However, the entities flowing through our simulation are not computer jobs but rather intelligent agents, namely users, engaging in their individual and social activities. As in ABMS, we will need to model their individual behaviors within a given social environment. Our agent models, however, can be greatly simplified since their learning or emergent behaviors are not our central concern.

Simulations of software systems are most effective when they are used to understand the important issues in achieving suitable performance or other operational characteristics. For us, this led to two sets of simulations to determine whether Escalier was workable and would likely provide utility. The first examined the basic operation of Escalier. Its outcome measures included the number of users needed to produce results that would be helpful to other users. Our second set examined the relative boost or decline in results from including expertise assessments. The second set of results will be compared with the first – was Escalier's performance enhanced with the use of expertise?

In general, evaluation of simulations can include validation through comparison with existing performance data. Unfortunately, systems under design rarely have such data. Instead, software system simulations are often evaluated largely on the credibility of their assumptions. We therefore detail our models and their assumptions in the following sections. Below, we discuss the problem space, outcome metrics, simulated environment, and simulated users.

3.8 Problem space and outcome metrics

To ground our simulations, we used the reduced system configuration problem discussed above. In this problem, we want to help users discover stable system configurations, but we simplify the system configuration problem to remove issues Escalier cannot handle (such as scripts). We argue that while this reduction limits the usefulness of Escalier to system configuration per se, it also allows us to examine the basic abstractions of Escalier use. We will reflect in the Limitations and Future Work section following our simulation results how Escalier might be used most effectively in conjunction with other software tools to handle system configuration, security, and activity traces. Meanwhile, we can model how Escalier functions.

To consider system configurations, we needed to divide the possible configurations C into two subsets: a subset of stable configurations C_s and a subset of unstable configurations, C_{ns}. Configurations are probabilistically assigned to one subset or another for each simulation run. In turn, since Escalier assesses whether a system configuration is stable or not, we will have true positives and false positives when Escalier has likelihood estimates that match the ground truth. We use as outcome metrics the number of true positives and false positives found by Escalier.

Note in the ensuing discussions, we will use the problem domain-specific term *stable* for the general term *valid* when it is appropriate.

3.9 Model of the environment

Given this problem space, determining a good model for the space of possible system configurations, and the stable and unstable configurations within that space, was the most difficult part of creating a good simulation.

We first considered laying out the system configuration space C as a grid, and uniformly randomly distributing the stable configurations in that grid. That was obviously not sophisticated enough.

We then examined an empirical dataset of 330 Linux configurations (the output of Linux's lshw –secure command), which we believe are all stable. Our analysis of the components of the Linux configurations suggested that, in general, they are likely clustered around one another: there was non-trivial duplication of large portions of settings. We found that an edit distance metric was useful in quantifying the statistical distance between a configuration's components (how far they are from each other). We calculated the pairwise edit distances between popular components of all configurations and aggregated the distances into a distribution. We discovered a positively-skewed log-normal distribution of edit distances (distances were mostly small), which gave us a topology that not only covered this dataset but could also generalize to a larger, synthetic dataset that used the same basic distributions. Specifically, we used the Hamming distance between configurations, which is a fixed string-length version of edit distance, because it was computationally beneficial to normalize the configurations in this way. We used the remaining components in our dataset to validate these findings.

Accordingly, we modeled the configuration space as stable configurations clustered around canonical ones. As mentioned previously, canonicals are configurations known to be valid. For the system configuration problem space, canonicals are system configurations known to be stable (e.g., tested by a vendor or open-source community). In a simulation run, canonicals were randomly placed in the space (i.e., the edit distances between them were random). Stable configurations were then created by randomly

selecting a canonical, randomly selecting a target edit distance from a log-normal distribution, and then randomly editing the canonical until it attained the target edit distance. Appendix B discusses in step-by-step detail how stable and canonical configurations are determined for the simulation.

Thus stable configurations are clustered around canonicals, and the ratio of stables to canonicals represents the *density* of the clustering. The *tightness* of the clustering is affected by the distribution of edit distances between the stables and canonicals (lower edit distances create tighter clusters). We also evaluated the effects of various cluster densities and different distributions to assess the robustness of the environment model.

We modeled a simulated configuration as a 50-parameter vector, where each parameter can have 10 values. The size of the simulation space C is therefore 10^{50} configurations. Each Monte Carlo run lays out between 1,020 (1,000 stables + 20 canonicals) and 20,300 (20,000 stables + 300 canonicals) configurations. The rest of the space consists of unstable configurations. This state space is relatively small for actual system configurations, but it is tractable for our simulations while also able to give us insights into how Escalier would work.

3.10 Model of user behavior

We needed a relatively limited model of users because we were primarily interested in their search behavior in the aggregate. White et al. (2009) show, in general, that there are *multiple levels of expertise* at play and that search behavior is affected by expertise level. Specifically, they found *experts were more successful* in their search sessions than non-experts, and experts tend to *explore longer* within their domain of

expertise by employing a *more broad and diverse search strategy* than do novices. Based on White et al., we modeled three characteristics of user behavior:

- *There are multiple levels of expertise*. Our user model consists of five levels of expertise ranging from relative novice to relative expert, which is based on Dreyfus' 5-stage model for skill acquisition (Dreyfus and Dreyfus 1980). The actual number of levels is not as important as the fact that there are multiple levels of expertise. By intuition, we assume higher levels of expertise are more rare than lower expertise. Thus a given user's expertise level is determined according to a Pareto distribution where novices are the majority and experts are rare.
- *Experts are more successful.* Users' levels of expertise correlate with their search accuracy within a search space. User accuracy is modeled by the shape of the distribution used to make edits to a configuration (see Appendix A for more details about user accuracy distributions). Experts modify their configurations using the same log-normal distribution of edit distances used to build the stable configurations, and novices use a "widened" log-normal so as to represent a near-equiprobable (i.e., random) selection of edit distances. Similarly, users with higher levels of expertise are more accurate in their reporting than novice users are.
- *Experts explore longer, more broadly, and more diversely.* Users' expertise levels also correlates with their level of effort in finding a stable system configuration. In other words, users with higher levels of expertise will

perform a more thorough search of the configuration space. In the simulation, experts make the most search attempts, and novices make the least.

Each user, according to her expertise level, searches for a stable system configuration. However, user assessments of stability are aligned with level of expertise (experts are correct in their assessment much more often than novices). Appendix A discusses in more detail how the user search and voting algorithm are determined for the simulation.

Note that while users in our model always act according to their expertise level, thus simulating a population of users, Escalier does not always know the expertise level. In the first set of simulations, Escalier cannot differentiate levels of expertise; in the second, it can use users' expertise levels to better its likelihood assessments.

Also note we assume that users only modify *existing* configurations. To "search" the space, users edit some number of settings within their own configuration. Therefore, edit distance is realistic for our model of how users move around in a system configuration space. We can directly measure distance as the number of edits needed to move from one configuration to another.

3.11 Simulations

Finally, our simulations used 10,000-round Monte Carlos (Mooney 1997) to investigate Escalier's usefulness under differing conditions. (Monte Carlos give an estimate of the true value of the outcome metrics, and 10,000 rounds provide a very tight variance for that estimate.) We used the Mersenne Twister random number generator with a period of $2^{19937}–1$.

Each Monte Carlo round randomly initialized the canonicals and the stables set. It then simulated the users, with randomly distributed expertise levels, using Escalier. Each user searched for a stable configuration, subject to his expertise level, and reported the results of that search to Escalier. Escalier then made a likelihood assessment of whether that reported configuration was stable. Each round simulated 200,000 users total.

We ran a 4x4x5x2 factorial simulation experiment where the factors were, in order, the number of users, the number of canonical configurations, the number of stable configurations initialized in the configuration space, and whether or not Escalier was able to consider user expertise. 200,000 simulated users was sufficiently large to assess potential utility at scale. In total, we ran 160 simulations, each with 10,000 rounds and 200,000 simulated users.

We remind the reader that, although we discuss our simulations in terms of configurations, the simulations use an abstract notion of a configuration: an integer tuple. Virtually any kind of information need can be represented in tuple form. A "stable configuration" means a tuple has correctly met an information need. Thus we believe the results of the simulation are applicable to the broader use case of social navigation.

3.12 Simulation Results

In short, our first set of simulations showed that Escalier worked in general. Table 1 shows a summary of the results for one set of parameters (20,000 stable configurations, 300 canonical configurations, expertise of the user is not used for likelihood assessments). This stimulation showed that:

- Even with a relatively small number of users for a social computing system (10,000 users), there are benefits: Escalier helped users discover about one-

third more stable configurations (36%) than were originally known. The gain in known stable configurations appears to grow dramatically (1168%) as more users interact with Escalier.

- By the time Escalier has seen 200,000 users, it has discovered over 3500 stable configurations, 18% of the actual stable configurations. Note this does not mean the crowd never reported on the other 82% of the stable system configurations – it means they did not do it enough for Escalier to decide that the configuration was probabilistically likely to be stable.
- The number of false positives is small, and the true positives that are found increase with the number of users.

Table 1. Escalier simulated performance using 20k stable configurations, 300 canonical configurations, and with expertise enabled. Values are means based on 10,000 runs.

# Users	True Positives	False Positives	% of Stable Space Found	Gain in Known Stables
10,000	108	0	1%	36%
50,000	1086	3	5%	362%
100,000	2092	12	10%	697%
200,000	3503	36	18%	1168%

All of these results were consistent across all simulations. The results were robust across all of the number of stable configurations used in our simulations, as well as the variation in the number of starting canonical configurations. Figure 5 shows there is positive growth in true positives as the number of users increase, regardless of the number of stable configurations in the simulation. We found similar results for the variation in number of starting canonical configurations: as users increase, so do the true positives, regardless of the number of canonicals.

In summary, based on the results of the first set of simulations, Escalier appears to find many more stable configurations than would be known without it. As well, the benefits scale as the number of users increases.

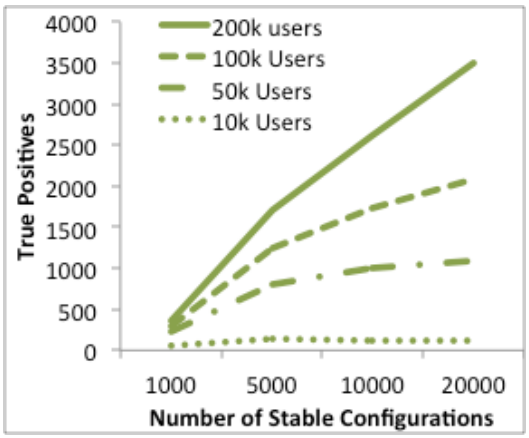


Figure 5. Effect of Number of Stable Configurations (300 canonical configurations; expertise disabled). The 4 lines are snapshots of number of users at that point in the simulation.

3.12.1 Using Expertise

The second set of simulations examined using expertise. In short, using expertise boosts the results. Escalier's ability to use a user's expertise seems to have a significant positive effect on helping users find stable system configurations more quickly than if expertise were not taken into account. , Figure 6 using the same parameters as Table 1 (20,000 stable configurations and 300 canonical configurations), shows the comparison between using expertise and not. Without using the expertise rankings for Escalier's likelihood assessments (i.e., if Expertise Layer were disabled), it takes about 160,000 users for Escalier to discover 3,000 true positives. When expertise assessment is enabled, it takes roughly 90,000 users, which is a 43% decrease in users needed. This boosting effect is especially valuable when there are relatively few users: At 10,000 users, using expertise is most effective, helping users discover three times as many stable

configurations when expertise is considered (339 true positives, excluding the canonicals) versus when it is not used in the likelihood assessments (108 true positives).

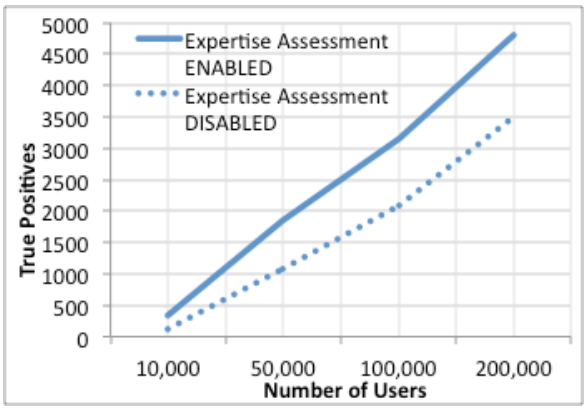


Figure 6. Effect of Expertise on True Positives (# of stable configurations found). This shows the effect of using Escalier's Expertise Layer's expertise assessments.

We also used additional simulations to compare the condition where the Expertise Layer only recognized three types of users. Figure 7 compares the effect of a discount approach against the original simulation using 20,000 stables, 300 canonicals, and with expertise assessments enabled. The figure shows there is little negative effect from collapsing the expertise rankings, and Escalier could use suitable discount expertise metrics.

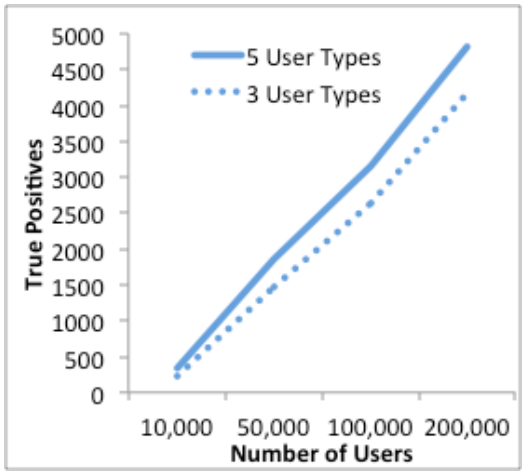


Figure 7. Effect of Discount Expertise Metrics on True Positives (20k stables, 300 canonicals, expertise enabled).

In summary, the simulations with the Expertise Layer being used showed that the use of expertise gives Escalier a noticeable boost, even when there are fewer users. Thus it is important to assess the users' expertise levels, if possible, for crowdsourcing systems like Escalier. They also suggest domain experts are between two and three times more effective at finding stable configurations than domain novices and about twice as effective as advanced beginners. Designing Escalier or similar crowdsourcing systems to incentivize or solicit input from domain experts is likely to be worthwhile.

3.12.2 Sensitivity Analyses

To evaluate the trustworthiness of our simulation results, we performed additional analyses on the assumptions in the models. We wanted to know how Escalier's performance would be affected if the assumptions in these models were changed. Although we looked at many of our assumptions, we report only three here.

We wanted to investigate the clustering density and tightness. One can vary clustering density by varying the ratio of stables to canonicals, and this was explored in the simulations presented above. With one exception, Escalier's effectiveness is reasonably robust to changes in cluster density. Escalier functions less effectively in either extremely sparse spaces with low numbers of users, or very dense spaces. It always returns true positives regardless of the density, and generally, as density increases, so do the number of true positives discovered. In the condition with relatively few users (10,000) and very sparsely filled spaces, Escalier finds relatively fewer true positives. This is not surprising since there are fewer searches, and each user’s search is less fruitful. As well, with 200,000 users in the most densely clustered condition, Escalier discovered 13% fewer true positives than the highest observed number of true positives.

In this condition, the space is so densely packed with stable configurations that users kept discovering the same ones instead of being forced to explore more of the space to find additional stable configurations. Therefore, if we were to change the assumption of how densely the space is clustered, we believe Escalier would maintain its effectiveness with a noticeable but acceptable decrease in effectiveness in the extreme cases.

Our environment model also assumed a tightly clustered space by using edit distances from a positively skewed log-normal distribution, which favors low edit distances. A normal distribution would cause larger edit distances to be used, and this would result in more loosely clustered spaces. To examine this, we ran a simulation using a normal distribution, and the results showed a decrease in true positives of almost 50% at 10,000 users and 36% at 200,000 users. Escalier does vary in its effectiveness if we were to change the assumption of the probability distribution for the edit distances or how tightly the configuration space is clustered. However, Escalier is only somewhat less effective in a more loosely clustered space, and it still helps users discover stable configurations at a significant rate.

Finally, we also found that the assessment threshold was reasonably robust. Reducing the threshold appears to only increase Escalier's finding of true positives, but at the cost of more false positives. Reducing the threshold by 25%, when there were 10,000 users, we see a 1.6x gain in true positives. The gain continues as users increase, although the effect decreases. However, the false positives also increase. With the decreased threshold and 200,000 users, the false positives triple, and when the threshold is decreased by 50% from the original, the false positives increase by an order of

magnitude. A reasonable threshold is likely between the original and the 25% reduced one, but it may be useful to dynamically change the threshold with the number of users.

In summary, even if the model assumptions were changed, all of the sensitivity analyses suggest that Escalier's performance would not deviate far from the findings in the original simulations.

3.12.3 Summary of Results

To summarize, under some basic assumptions, the study of Escalier's simulated performance argued that Escalier would work as claimed and:

- In the simulations, Escalier scaled appropriately. As the number of users increases, so did the number of true positives found.
- We did not have additional "noise" as Escalier scaled. We found it was possible to tune Escalier to keep the number of false positives low as the system scaled.
- Knowing users' expertise levels boosted Escalier's results, substantially in most cases. However, knowing users' expertise levels was not required.
- Discount expertise measures are likely to be effective, and difficult-to-obtain measures are likely not to be necessary.

In short, under reasonable assumptions and in reasonable conditions, Escalier appears to be a workable crowdsourcing system, and the use of expertise improves its effectiveness.

3.13 Limitations

Our results are limited in that simulation studies provide evidence of workability and can examine the possible effects of system changes, but they are not empirically-derived proof. Simulations also cannot substitute for field studies where emergent behavior and everyday messiness can be observed. Nonetheless, we believe the simulations were useful for the understanding we required.

As well, our use case scenario was necessarily reduced. For example, Chau et al.'s Polonium (2011) is more tailored to configuration management, and, in general, system configuration systems are substantially better for that particular domain. We remind the reader, however, that we were using the use case to motivate and then understand Escalier's use of expertise.

We also see further examining the use of experts in systems like Escalier. Proper incentives for experts to explore more configurations could further bootstrap Escalier's true positives. Alternatively, users might also want to see the evolution of experts' configurations over time. Curious users could then follow the traces of configuration maintenance.

We believe that expertise can be used more within Escalier or its applications. The system could be able to ask domain experts to reconcile differences or add rules, for example, when inconsistencies are inferred.

3.14 Conclusion

We began by claiming three contributions for this chapter. They were (1) using expertise could substantially benefit crowdsourcing systems, at least for some applications, (2) Escalier was an innovative crowdsourcing system that used traces and

configurations from the crowd, and (3) simulations had utility as a way of testing the workability and feasibility of social computing systems like Escalier. We believe we have demonstrated each of these. In this chapter, for these contributions:

- We presented Escalier and Escalier applications for an example use scenario to show how it worked.
- We presented how Escalier could use the expertise ratings of users by weighting user reports to affect its likelihood assessments of specific configurations.
- As a result of use, both users and a community have mutually-reinforcing motivations to use the system. Users can find out whether they have configurations likely to be stable and in return, the community gains a repository of suitable configurations.
- We also showed that Escalier can operate effectively out of the box. It can bootstrap using canonical, or known, configurations, thus avoiding the cold start problem.
- We showed in a simulation study of Escalier that it can provide utility to users by finding new configurations, and that utility scales appropriately with the number of users.
- We also showed that the use of expertise boosted Escalier performance under the simulation conditions.
- Finally, we showed how simulations were able to study Escalier at scale for basic workability and feasibility.

In this chapter, we have described the Escalier system, and through a study of its feasibility at scale, we have answered the first research question posed in this thesis:

- **RQ1**. Under what conditions, and to what extent, does mixed expertise within a crowd materially benefit a crowdsourcing system at scale?

Using Escalier, we investigated expertise at scale for fact-based expertise criteria, and there were some feedback loops (i.e., user reports and external expertise assessments) and mixed expertise leveraged from the same crowd. We wanted to do a deeper investigation of feedback loops and mixed expertise in different crowds, particularly in a domain where opinion-based expertise criteria was at play, so we built and studied another system with those motivations in mind. Using the same expertise layering principles we used for Escalier, we built the Kurator system. The next chapter explains Kurator and how it answers our second research question.

Chapter 4. Kurator: Mixed Expertise between Crowds

The previous chapter's study of Escalier taught us that mixed expertise within a crowd materially benefits a crowdsourcing system at scale, and that feedback loops (i.e., user reports and external expertise assessments) were helpful to the system. We now know, in theory, that leveraging a range of expertise levels in the crowd is worthwhile, particularly where the diverse expertise of a population can be leveraged to uncover more of a solution space. Using what we learned, we wanted to establish, *empirically*, if mixed expertise was beneficial, and in what other problem areas it had benefit. We also wanted to study more fully the effects of feedback loops in a crowdsourcing system.

In this chapter, we present the Kurator system, which expands the pool of mixed expertise contributors to include an expert crowd as well as machine learning agents. Kurator is designed to allow the expert crowd to perform work alongside, and to provide feedback to, a paid crowd and machine learning agents. This mix of expertise between crowds allows us to address the second research question:

- **RQ2**. Under what conditions, and to what extent, is there benefit when using a mix of crowds, differentiated by types and levels of expertise, to solve a problem when the crowds work on similar tasks?

Kurator is a solution within the second problem area discussed in Chapter 1, which focuses on personalized, subjective problems, where there are multiple valid

solutions to an end user. Thus Kurator uses opinion-based criteria (see Chapter 2) for measuring expertise in predicting a user's subjective preferences. We built Kurator using the principles of expertise layering to demonstrate the utility of this paradigm in an empirical study.

4.1 Introduction

People have much more digital content than they can manage, even when it comes to relatively narrow sub-sets of content, such as digital photos. Researchers have called out the need for a strategy for forgetting, preserving, and remembering personal digital content (e.g., Nejdl and Niederee 2015), but this requires families to significantly shift their habits, which is often impractical. Instead, a focus on using systems to help manage familial digital information may be valuable (Gulotta et al. 2015). Since artificial intelligence (AI) is not able to make judgments about personalized, subjective content, approaches like crowdsourcing look promising. However, most prior crowdsourcing work has focused on inferring specific preferences by finding similar individuals or groups in the crowd (similar to collaborative filtering), or on solving general problems where everyone has some amount of the required expertise in the problem domain – neither of which are likely to be the case with personal data.

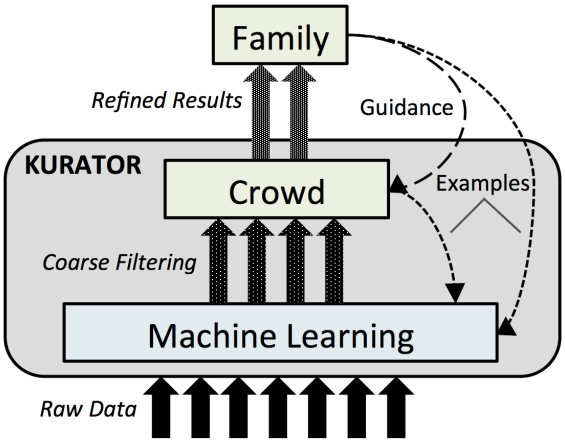


Figure 8. Kurator system diagram. Kurator starts with a collection of digital media content. A machine learning tier reduces the amount of content by filtering, based on criteria for that media type (such as no volume for audio). The crowd tier then does further refinement, producing a candidate set for the family, who is the ultimate judge for family memory. Feedback from the family can guide the improvement of the machine learning tier and the crowd tier.

This chapter introduces a novel approach that uses mixed expertise crowds as part of a hybrid intelligence system to reduce the curation burden on families. We explore this approach via Kurator, a system designed to help families curate their own digital audio recordings. Kurator is a hybrid intelligence system (see Figure 8) because it uses inputs from machine learning and crowds. It also leverages the mixed expertise levels between "crowds": families (experts) and paid web workers (variable expertise). We implemented a tiered refinement approach whereby a machine learning classifier performs coarse-grained filtering on a family's entire digital audio collection, and the crowd refines the classifiers' output into a smaller, more manageable set of higher quality recordings that can be presented back to the family. Kurator also uses feedback from families and paid crowds as a way to obtain more personalized results by providing feedback to the ML classifier and the paid crowd. Providing feedback to this hybrid intelligence system can improve the system's ability to return more accurate,

personalized results for families. We evaluated Kurator through a user study with five families.

We found that not only is the resulting curation useful but also that crowdsourcing can be applicable to a class of problems we would not expect. Curating families' digital memories belongs to a class of problems where it is not obvious the crowd can help: those for which specialized knowledge is needed (i.e., a family's preferences), and the crowd is not the target audience (the family is). For these problems, we would not necessarily expect crowdsourcing to be a viable solution, but as we will show, Kurator is able to effectively leverage crowds to provide useful assistance. We believe our work demonstrates a potentially important new problem setting in which crowds can benefit users.

We make the following contributions in this chapter:

- Through the problem of digital curation, we show there is a class of subjective problems where crowds may be helpful but have not yet been leveraged. Specifically, we show that crowds are effective at predicting whether specific children's digital audio recordings will be valuable family memorabilia.
- We introduce the Kurator system, a hybrid intelligence system which uses mixed-expertise crowds in a tiered architecture to synthesize inputs from multiple layers of contributors, such as machine learning, the crowd, and the family, to reduce the burden on a family with family curation tasks.
- We show that Kurator can ameliorate the digital curation problem as well as the general validity of our approach through a user study, as well as a set of crowd experiments.

After we walk through the background literature for our work, we then present Kurator and its major design assumptions and features. We then present the results of our user study and laboratory experiments where we examined whether Kurator and its features were effective. We conclude with a brief discussion and limitations.

4.2 Background

4.2.1 Curation in HCI and CSCW

The practices and processes of selecting, organizing, and maintaining a collection of material is broadly considered as "curation." Curation has been extensively studied in institutional archives and library science, and it has recently been extended to data and digital curation (Yakel, 2007). The problem of curating personal digital content is a difficult one that remains difficult and unsolved. Marshall (2007) noted that "digital material accumulates quickly, obscuring those items that have long-term value (p. 5)." Marshall (2007), as well as Marshall et al. (2006), found that almost all users do not do an adequate job of curation.

Early work on digital curation in CSCW/HCI focused on studies of and systems for sharing digital photos. Recent curation research in CSCW/HCI has focused on the work of curation in social media sites. Chang et al. (2014) examined the curation work taking place a social curation site, Pinterest. Zhao and Lindley (2014) examined how the use of a social media site leads to a curated archive of personal digital content. Very recently, there have been studies of people's perceptions and understandings of algorithmic curation on Facebook News Feed and how it affects their use of the system (Rader and Gray 2015, Eslami et al. 2015). As well, a recent study on the "modern day baby book" investigated new mothers' photo sharing activities on Facebook (Kumar and

Schoenebeck 2015). These studies did not investigate how families might curate digital content when they do not want to share or keep private.

4.2.2 Automating personal digital media curation

It is clear that people can hand select digital content for preservation and use. Relatively little work has investigated how digital curation might be done through systems, either machine learning-based or crowd-sourced.

Obrador et al. (2010) inferred user preferences for "style" using social cues from their online photo albums, but it did not allow for explicit end user feedback into the system. Other work that builds on the idea of inferring user preferences includes Guldogan et al. (2013), which required a profiling task to be performed by the end user. This, however, required training, and a usability goal might be to be effective "out of the box" without requiring user tasks before being useful, as we do with Kurator.

Recently, there has been research on automating personal digital media curation using general preferences for photo selections instead of user-specific preferences (Ceroni et al., 2015a, Ceroni et al. 2015b, Nejdl and Niederee 2015). Nejdl and Niederee (2015) concluded that a coverage-based approach, which attempted to cover multiple events, did not perform as well as a simple reduction-oriented strategy, which removed duplicate and near-duplicate photos. We follow this, using a similar reduction-oriented strategy in Kurator by filtering out low quality recordings. Additionally, this line of work suggests a potential utility in utilizing family-specific preferences by re-training the machine-learning agents on data supplied by the family or the crowd.

The only direct example of crowdsourced curation of personal digital media, Cusano and Santini (2014), proposed a community-sourced method to help users

categorize their photos using labels (i.e., tags). This method correlates photos from the public Flickr user community with target users. Public photos with labels are used to predict labels for similar photos from a target user. This method works only when there is an Internet-scale repository of public data, and is appropriate only with some content. Kurator is designed to work when there is no public repository of similar data.

As an indirect example of crowdsourced curation, Organisiak et al. (2014) used profiling tasks to understand user preferences, then they employ two approaches to understand a user: taste-matching and taste-grokking. Taste-matching works by finding workers similar to the user's profiling results, and with taste-grokking, where any benefit is limited to when the users train the crowd. As mentioned, we want an approach that is effective immediately but also improves with additional training.

Similar to taste-grokking, Yi et al. (2013) leveraged a user's response to pairwise comparisons from a subset of items. They use a matrix completion algorithm, which they call crowdrank, to infer the user's preferences on the remaining items. This matrix completion approach, however, can be very lengthy, increasing the task time and cost significantly.

4.2.3 Using expertise in crowdsourcing

Our work also overlaps with systems leveraging collaboration between only expert crowd workers, between experts and non-experts, and among some mix of expertise. In Chapter 2 of this thesis, we reviewed the literature on the use of expertise in crowdsourcing, and we drew attention to these types of systems. Kurator builds upon that work by combining the mutual efforts of the crowd, expert users, and machine learning agents, in addition to leveraging expert users' feedback. As well, Kurator's mix

of contributors represents a mix of expertise between "crowds," and Kurator uses a greater diversity and depth of mixed expertise than what we have seen in prior work.

As a point of emphasis, we remind the reader that by following the principles of expertise layering, we designed Kurator to work without *requiring* the expert crowd's involvement, especially when the system initializes with a new family. This subtle but important point is what makes Kurator distinct from the prior literature, where experts are used to guide non-expert crowds but not contribute to the same tasks directly (e.g., Dow et al. 2012), or in a division of labor where experts work on different tasks than the non-experts because those tasks can only be done by experts (e.g., Huang et al., 2015).

4.3 KidKeeper Background

Kurator currently uses digital audio recordings collected from the KidKeeper system (Jones et al., 2016). KidKeeper is a toy-like device designed for children to spontaneously capture audio recordings of their everyday activities, combined with a simple curation and delivery system to enable parents to enjoy the recordings their children created. The types of content captured using KidKeeper are children singing, telling a story, making up sounds or words, screaming, and short phrases. KidKeeper revealed the need for a more sophisticated, automated curation system to help parents find the "gems" of audio recordings (Oleksik and Brown 2008) from a large digital audio collection. Next, we explain how Kurator addresses this issue.

4.4 Kurator

Improving personal digital content curation requires trading off two key factors: scalability and access to specialized knowledge. A family has "expert" knowledge of

what is meaningful to them, but their time is finite resource. Machines can scale to massive data sets, but cannot understand the "meaning" of content, making only superficial assessment possible. Crowds of online workers are flexible, available on demand, and can be recruited at scale. Furthermore, crowd workers will likely have some level of common social understanding with the family. But the crowd is still separate from the family and does not know the subtler context underlying the content. Additionally, crowdsourcing can often be cost-prohibitive for very large collections.

Kurator is a hybrid intelligence system that reduces the time and effort cost of curation for families so as to make collections of digital memories easier to manage. It uses a tiered architecture (see Figure 8) that first filters raw data using machine learning, and then asks the crowd to assess the content on behalf of the family. Finally, the filtered, significantly smaller set of potentially-interesting artifacts are returned to the user for final evaluation. After viewing and (optionally) further refining the set, family members can provide feedback to the crowd and machine learning to improve future results. We apply Kurator to the domain of personal digital audio recordings collected using KidKeeper.

4.4.1 Example Scenario

Daniel, hearing the sound of young children running down the hallway of his hotel room, gets a twinge of nostalgia for his own children. He logs onto the Kurator website to listen to some audio recordings of his kids. He notices two things right away. First, he see there are now over 1,000 recordings in his collection, and a part of him is thankful he hasn't listened to the vast majority of them. The other thing he notices is that his Top 20 list has three recent additions. He listens to the first recording and,

enjoying his son's rendition of *Hush Little Baby*, tags and rates the recording accordingly. He enjoys the second recording, of his daughter saying how much she loves her daddy, and tags and rates it. The third recording, the longest of the three, is less enjoyable because the family dog is barking for half of it. He clicks on the feedback link for this recording, and on the subsequent page, he sees all the previous guidance he and his wife have provided up until now. Seeing that they had, somehow, not yet provided guidance about their dog, Daniel submits the following feedback to the system: "It's not as meaningful if the dog is barking for too much of the recording."

4.4.2 Design Considerations

Kurator is designed to address the fact that there is no way for (most) people to keep up with their digital media collections long term. We make a few baseline assumptions about curating digital artifacts:

- *Curation is a process and not a static goal*. It is dynamic over time as tastes, goals, needs, and perspectives change.
- *Everything should be kept*. Digital space is cheap, so curation should no longer be about "keep or throw away" but rather about "what to pay attention to".
- *The primary goal of curation is not to select the single most meaningful artifact*. Even families themselves may not be able to do this. Instead, the goal is to narrow the focus down to a meaningful *set* of artifacts for further processing by the family.

Below, we discuss Kurator's key design features: the integration of machine learning, the crowd, and families, and how we incorporate feedback in this process in order to improve results.

4.4.3 Integrating machine learning

Design Rationale: We leverage machine learning to reduce the decision space for the human contributors. Reducing the curation decision space by using automated approaches in a reduction-oriented strategy has been demonstrated on digital photo collections (Nejdl and Niederee 2015). The automated approach we use needs to handle continually re-training machine learning classifiers over time as the crowd and the family provide inputs to Kurator. For this purpose, Nguyen et al. (2015) suggest using logistic regression with gradient descent, which supports incremental training.

System Description: Kurator uses a three-class rating system, where each audio recording is rated as one of three classes. Thus the machine learning classifier (ML) is currently implemented as a multinomial, or multi-class, logistic regression model using gradient descent. This particular ML is meant as a proof of concept, and Kurator is designed to be agnostic to the ML algorithm and even to the use of an ensemble, or a "crowd", of ML algorithms. The core of the ML is implemented in Octave (Eaton 2009) scripts, which are called from a Python script using the oct2py module. When Kurator is initialized for a family, there are no human ratings to use to train the model, so regression coefficients from a preliminary study are used as the seed. In practice, regression coefficients could be reused from other families who have already used the system. The ML is re-trained as human ratings become available, and as new artifacts are uploaded to Kurator, the ML predicts ratings for them. Also, our purpose for the ML is to remove

low quality artifacts, in a reduction-oriented strategy, because low quality audio recordings likely have more objective characteristics (e.g., noisy or blank recording).

For feature selection, we analyzed the KidKeeper data set. The most common recordings captured with the KidKeeper device were *songs*, *stories*, and *screaming gibberish*. In order to characterize these types recordings, we used these general principles:

P.1 Screaming produces higher average amplitude than talking.

P.2 Singing or talking has more frequent and dramatic changes in amplitude than constant screaming, random noise, or a blank recording.

P.3 Longer recordings contain more content and are therefore more likely to have interesting content.

The features currently implemented are root mean square (RMS) of the spectrogram (addresses P1), RMS of the peaks in the spectrogram (addresses P2), duration of audio (addresses P3), and ratio of the peaks to the raw RMS (addresses P1 and P2).

Note that our aim is not to create a state of the art machine learning classifier (ML) to eventually replace humans in the loop. We deliberately left out a feature that we thought would be needed--detecting adult voices. There are also many other speech classification features and methods we did not incorporate, such as emotion detection (e.g., Schuller et al. 2011 and Le and Mower Provost 2013), speech activity detection (e.g., Sadjadi and Hansen 2013), and age and gender detection (e.g., Meinedo and Trancoso 2011 and Hämäläinen et al., 2014). Because the problem of *personal* digital content curation is highly subjective, we assume the ML is limited and will eventually

fail on some content, no matter how sophisticated the ML is. We wanted to know whether the family could guide the crowd where the ML failed. With our current implementation, we can test this easily and determine if Kurator is robust to a limited ML. Future implementations can use more sophisticated ML mechanisms.

4.4.4 Integrating crowd input

Design Rationale: Clearly, machine learning has limits, particularly on highly subjective tasks like personal digital curation, where "personal importance" is a key criterion for users in their decision-making (Ceroni et al., 2015a). As discussed above, crowdsourcing has been used on subjective tasks and in personal digital media curation directly (Cusano and Santini 2014) and indirectly (Organisiak et al., 2014 and Yi et al., 2013). Thus harnessing the power of crowdsourcing seemed to be a promising approach to consider for this problem.

System Description: Our prototype implementation of Kurator uses Mechanical Turk as its generic crowd. It also uses Amazon's Simple Storage Service (S3) to store the audio files, making them read-accessible only for the duration of crowd tasks. Crowd tasks are automatically generated by Python scripts using Boto3, a Python interface to Amazon Web Services, to allow for API access to Mechanical Turk and S3. We built a crowd-tasking engine to interface with Mechanical Turk. This engine automates the workflow of creating HITs, collecting responses, and tracking the accuracy of workers' responses. A HIT consists of a description of the task, a link to the audio file, a subjective scoring section, and a free-text feedback section.

We used, as the description of the task, the question: "*Do you think this audio could be meaningful to the content owner?*" Workers were given three options

("Definitely", "Maybe", "No Way") as well as a free-text feedback section to answer the question: "*Why did you rate it that way?*" We allowed three workers per HIT and used majority voting to determine the crowd's rating on a particular audio recording. Furthermore, the crowd's ratings were later used to re-train the machine learning classifier.

4.4.4.1 Are we asking the right question?

A key issue with integrating the crowd's input was that we did not know how to elicit crowd responses that were in line with the parent's responses. We investigated the effect of changing the wording of the crowd task question altogether. We tested four questions on the same 30 audio recordings where we had ground truth data, and we prevented crowd workers from working on more than one question. The questions were:

A. *"Do you think this audio could be meaningful to the content owner?"*
B. *"Do you think the content owner would want to hear this again in the future?"*
C. *"Would you want to hear this again in the future?"*
D. *"If this were your child, would you want to hear this again in the future?"*

Quantitatively, we found that Question A's ratings were the only ones with at least moderate agreement with the ground truth (kappa > 0.4). Question B showed fair agreement (kappa > 0.3), which suggests that framing the question as a judgment about the content owner*'s* preference might elicit crowd responses that are in line with the parent's responses.

As confirmatory evidence of this, for Question C, there were many negative remarks in the "*Why did you rate it that way?*" feedback, such as "I'm not sure why I would want to listen to a toddler singing if it were not my child. This recording is

moderately creepy", and "This is grating on the ears." As well, crowd workers used the word "annoying" much more often to describe a recording when answering Questions C or D.

4.4.5 Integrating the family's expertise

Design Rationale: The crowd inherently does not have as much situated understanding as do family members. Therefore the family should add its expertise, but only in a cost-effective manner for them. Our goal was to allow family members to rate only the content that had been deemed possibly appropriate by the machine learner and the crowd.

System Description: We implemented a family-facing website built with Django and Bootstrap. Referring back to Daniel's activities in the scenario above, he interacts with the website to access his family's personal content. As he listens to recordings, he provides ratings (similar to how the crowd provides ratings) and occasionally submits keyword tags.

To solicit ratings and text tags from the family, every audio playback web page contains: a question--"Would you want to hear this again in the future?", clickable buttons to answer the question – "Definitely", "Maybe", and "No Way", and a free-text area to submit keyword tags, limited to 140 characters (same as Twitter). The three categories of ratings are the same for the family and the crowd.

Note that Kurator is designed to incorporate family-sourcing as well, where multiple parents, other family members, and family friends can be included--technically, anyone the parent wants to give access to. Each person would have his or her own login, and the parent can restrict whom to give feedback access to. Intuitively, close family and

friends probably have more knowledge about the family than a generic crowd does, so we expect the family "crowd" to have higher levels of expertise than the generic paid crowd.

4.4.6 Implementing feedback loops

Design Rationale: Since the family is the end user of Kurator, their subjective preferences need to be considered. As they are the experts for this task, it could benefit Kurator to allow the family to "shepherd" the crowd (Dow et al., 2012) by being the source of external feedback to crowd workers. Natural language descriptions have proven to be an effective way to guide crowd workers (e.g., Dow et al., 2012, Zhang et al. 2012 and Kokkalis et al. 2013).

System Description: The Kurator website allows the user to provide guidance to the crowd. Family guidance is used verbatim in the tasks assigned to the crowds. Furthermore, the family's ratings, as well as the crowd's ratings, are used to re-train the machine learning classifier.

4.4.7 Summary

Kurator is a hybrid intelligence system that uses machine learning along side mixed-expertise input from people (crowd workers and families) to weed out low quality artifacts. As Kurator's tiered process moves from machine learning to the experts, the task requirements are increasingly subjective. The use of machine learning, crowd workers, and experts to collectively label items has been used in active learning (Nguyen et al., 2015), where the aim is to minimize costs over a finite pool of items. Kurator differs in that the pool of items is not finite (i.e., the pool of artifacts grows over time), and due to the subjective nature of the personal digital curation, the "ground truth" (what a family deems important) can vary.

4.5 User Study and Experiments

To better understand if Kurator's tiered architecture helped reduce the level of work for end users, we ran a user study with five families. To explore how guiding the crowd and re-training the ML improved system performance, we ran a series of focused follow-up experiments.

4.5.1 Study Design

The study participants were five families who had used the KidKeeper system for about a week to capture recordings of their children. The study was designed to obtain ground truth ratings as well as qualitative data from semi-structured interviews. The ratings were the categorical responses discussed above: "Definitely", "Maybe", and "No Way". Parents also chose their absolute favorites from their list of "Definitely"-rated recordings, with no minimum or maximum number suggested or required. We did this in order to evaluate how well Kurator could find a parent's favorite recordings, which we refer to as Favorites throughout the rest of the chapter. The interview consisted of questions about parents' decision-making processes for their ratings, the difficulty of doing the ratings, what they thought about Kurator's selected recordings, and if they could train someone else to recognize their preferences.

An issue we had to overcome was the size of each family's audio collections: they ranged from 217 to 620 recordings. The limiting factor for the user study was the parents' time, and we felt that having parents rate their whole collections would have been unreasonably burdensome. Thus we used a randomly sampled subset (n=120) for each family.

4.5.1.1 Sampling

In a preliminary study with a different family, we found that simple random sampling resulted in too many low quality audio recordings (i.e., those rated as "No Way"). After two researchers coded all study participants' recordings into the three categories, we found there were a much larger percentage of non-keepers (No-Way's) than keepers (either Maybe's or Definitely's). Accordingly, we used a stratified non-proportional random sample (Bernard, 2011) of the coded recordings to equalize the number of recordings in each category (n=40 for each of three categories). For the rating task in the user study, the Kurator website was configured to present one recording at a time to the user, and each recording was randomly sampled from the sampling frame.

4.5.1.2 Baseline ML parameters

We needed to seed the machine learning classifier with a baseline set of parameters, so we trained the ML on rating data collected in a pre-study. Kurator used the same baseline for each family.

4.5.1.3 Kurator's Top K

The study included a comparison between the parent's Defintiely's (including Favorites) and Kurator's selection for what it thinks the top *k* recordings are. We set *k=12* (10% of 120) and played at least that many recordings during the interview.

4.6 Findings

In this section, we use the ground truth data collected from the user study in follow-on experiments, as well as interview data. In our evaluation, we use precision,

recall, and F1 scores to measure Kurator's overall performance as well as the performance of its hybrid intelligence components: the crowd and the ML classifier.

4.6.1 Kurator worked

In the interview data, there were two sets of preferences that parents followed. These were not always mutually exclusive, as they were preferences.

Overall, parents viewed Kurator as a tool to augment their curation work by reducing the overall workload. We called this preference "Best-Of" because the user wanted to hand-curate a reduced set. One parent remarked: *"I don't even go back and look at all 60,000 pictures that I have on my computer. If it's going to send me a smaller sample, I'm more likely to listen to all of them."*

Another parent also elaborated on the benefits of working on a reduced collection: *"[Maybe if] it saved 10 minutes worth of samples, where it's small enough that you could sit down and kind of click through them quickly and figure out if you like it or not. You got time for that in between other stuff, where if it's large, large number of stuff, when are you going to sit down and actually go through it?"*

A third parent acknowledged her use for Kurator would depend on the frequency of her curation efforts: *"I would probably let [it] give me the top 20. If I knew this was going to happen once a week, I would let it do it for me. Yeah, I think I would just definitely choose the efficiency over making sure I captured every single moment."*

Table 2. ML classifier's precision, recall, and F1 scores.

Precision - Machine Learning							
		Participants					
		A	*B*	*C*	*D*	*E*	*Total*
Rating	Definitely	-	1.00	-	-	-	0.43
	Maybe	0.33	0.07	0.04	0.19	0.09	0.14
	NoWay	0.70	0.56	0.94	1.00	1.00	0.76
Recall - Machine Learning							
		Participants					
		A	*B*	*C*	*D*	*E*	*Total*
Rating	Definitely	-	0.05	-	-	-	0.03
	Maybe	0.65	0.50	0.75	0.95	1.00	0.76
	NoWay	0.70	0.70	0.31	0.18	0.28	0.38
F1 Scores - Machine Learning Classifier							
		Participants					
		A	*B*	*C*	*D*	*E*	*Total*
Rating	Definitely	-	0.09	-	-	-	0.05
	Maybe	0.44	0.13	0.07	0.32	0.16	0.24
	NoWay	0.70	0.62	0.47	0.30	0.43	0.51

Table 3. Crowd's precision, recall, and F1 scores.

Precision - Crowd							
		Participants					
		A	*B*	*C*	*D*	*E*	*Total*
Rating	Definitely	0.58	0.91	-	0.46	0.14	0.58
	Maybe	0.39	0.08	0.05	0.26	0.08	0.17
	NoWay	0.87	0.83	0.94	0.86	0.96	0.89
Recall - Crowd							
		Participants					
		A	*B*	*C*	*D*	*E*	*Total*
Rating	Definitely	0.38	0.52	-	0.43	0.67	0.43
	Maybe	0.61	0.38	0.75	0.52	0.50	0.56
	NoWay	0.75	0.80	0.48	0.66	0.50	0.60
F1 Scores - Crowd							
		Participants					
		A	*B*	*C*	*D*	*E*	*Total*
Rating	Definitely	0.46	0.66	-	0.44	0.24	0.50
	Maybe	0.48	0.13	0.10	0.35	0.14	0.26
	NoWay	0.80	0.82	0.64	0.75	0.65	0.72

Table 4. Kurator's precision, recall, and F1 scores.

Precision - Kurator							
		Participants					
		A	*B*	*C*	*D*	*E*	*Total*
Rating	Definitely	0.64	0.95	-	0.46	0.15	0.55
	Maybe	0.42	0.09	0.04	0.26	0.09	0.17
	NoWay	0.71	0.59	0.92	0.86	0.97	0.80
Recall - Kurator							
		Participants					
		A	*B*	*C*	*D*	*E*	*Total*
Rating	Definitely	0.31	0.31	-	0.43	0.67	0.31
	Maybe	0.45	0.25	0.50	0.52	0.50	0.46
	NoWay	0.87	0.92	0.57	0.66	0.54	0.67
F1 Scores - Kurator							
		Participants					
		A	*B*	*C*	*D*	*E*	*Total*
Rating	Definitely	0.42	0.46	-	0.44	0.25	0.39
	Maybe	0.44	0.13	0.08	0.35	0.15	0.25
	NoWay	0.78	0.72	0.70	0.75	0.69	0.73

These sentiments about Kurator's utility as a curation tool is further supported by our quantitative data. Kurator was effective in refining the families' collections by systematically removing non-keeper recordings. **Table 4** shows that Kurator had 80% precision, 67% recall, and a 0.73 F1 when predicting NoWay audio clips. This suggests the tiered refinement approach may be a reliable way to winnow a collection down simply by removing artifacts of the lowest quality. In terms of raw numbers, Kurator removed 342 out of 600 audio clips (120 x 5 families), and it did so with 78% precision, meaning for every four recordings the system filtered out, three of them were truly non-keepers. This metric, alone, suggests Kurator may be a useful tool for reducing the workload on families who prefer to hand-curate a reduced set to find the "Best-Of." We will discuss this more below.

At other times, parents talked about finding "gems" in the collection. These are essentially a set of clips in an equivalence class. In this preference, parents were less interested in winnowing down their collection so they could hand-select interesting clips. Instead, they were happy when Kurator found "gems" even if it didn't find all of them. We called this preference "Album", and it was manifested as a desire to hear many sufficiently interesting recordings instead of only a few very meaningful ones. As an example, one parent enjoyed when Kurator returned an audio clip of his young boy reciting the following line from the movie The Princess Bride: *"My name is Inigo Montoya. You killed my father. Prepare to die."* Another parent expressed using a low threshold for what she would find interesting enough to listen to again: "*I pretty much would definitely listen to all the ones that weren't garbage files. I liked all of the ones that were of them talking...*"

The following sections discuss how each tier of Kurator worked and then goes into more detail about how well the system worked overall.

4.6.2 ML is effective at filtering non-keepers

As the first tier of the tiered architecture, we want to know if the ML was effective in terms of its quality of predictions. **Table 2** shows the ML's effectiveness in terms of precision, recall, and F1 scores. Overall, the machine learning classifier had an F1 score of 0.51 in finding non-keepers (NoWay's). For the three most selective families (i.e., they rated the least number of clips as Definitely's), the ML had 100% precision for two of them and 94% for the other, when predicting No-Way ratings. This is likely due to these families strongly favoring the "Best-Of" approach, which means they tended to rate a large majority of their collections as NoWay. This would increase the likelihood that a ML-rated NoWay was also rated as NoWay by the family, thus driving up the precision for NoWay's.

4.6.3 The crowd is effective

As the second tier, the crowd was overall particularly good at identifying non-keepers (NoWay ratings), achieving 89% precision and 60% recall (F1=0.72) across all families combined (see **Table 3**). The crowd was only moderately successful in predicting Definitely ratings (58% precision, 43% recall, 0.50 F1), but for one family, the crowd achieved 91% precision (n=35). This family rated significantly more Definitely's (n=62) than the other 4 families, which would drive up the crowd's precision for Definitely's, similar to the previous discussion about the ML's high precision for NoWay's.

Note that the recall for four of the families ranged from 38% to 67%, meaning the crowd was able to uncover a significant subset of the Definitely's, in general. The recall and precision of zero for Family C's Definitely's may have been a consequence of their using nuanced and idiosyncratic criteria, which we discuss below.

Interestingly, four crowd workers expressed their enjoyment, via unsolicited emails to the research team, in doing the task of rating audio clips of children. Two workers said they "loved" hearing these clips, commenting on the cuteness and hilariousness of the children's utterances. One worker even remarked: *"As mine grow up I wish I had saved so much more audio of them."*

Remarkably, the crowd divulged an interesting array of thought processes and criteria they used to make their decisions in their free-text responses in the tasks. Beyond frequent statements about recordings being "cute", "silly", and "adorable", workers often viewed specific activities they heard as being important to parents, such as singing, playing, and a *"child calling for her daddy...means so much"*. Some workers guessed about possible use cases that would make an audio clip valuable, such as: *"meaningful...if long distance"* or *"to a parent who isn't around at the time this occurred"*, *"put into a musical Christmas card...sent overseas if they have a parent in the military"*, and *"they might want to embarrass their kid when he's older; quite funny"*. Others made judgments, different from their own opinion, based on what they thought the parent would choose: *"it's a child screaming, perhaps [the parent] thinks it's funny but it's annoying really"* and *"I think this audio will only be meaningful to the audio owner...while cute, it doesn't mean a lot to people who do not know the child or have some context to go with audio."* Finally, many workers were willing to share personal

thoughts about the audio recordings themselves: *"Reminds me of my kids", "heart breaking child wishing for parents, so moving", "sounded awesome; [my] favorite so far", "I love kids just being kids", and "children grow up so fast".*

4.6.3.1 Identifying specialized crowds

Given the crowd's apparent ability to draw on their own experience or to use in-depth thought processes to predict ratings for others, we wanted to better understand if there was a subset of the paid crowd that seemed to perform better than the rest of the crowd. To identify a "specialized" crowd, we ran an experiment where 40 crowd workers rated 30 randomly chosen recordings from two families (15 from each family). In addition to soliciting the ratings, we asked demographics questions of the workers: their age range, gender, how long they have been a parent, and how many children they have.

We calculated each worker's accuracy based on how their ratings compared to the ground truth (i.e., the parents from the user study). Of those who have children (n=16), workers 35 and younger rate less accurately than those who are 36 and older (47.8% vs. 63.3%, respectively; Fisher's exact test p-value=0.02). Remarkably, when comparing the same age groups for workers *without* children, we see an opposite effect: workers 35 and younger were more accurate (55.8%) than those who were 36 and older (48.0%), but the effect was not statistically significant (Fisher's exact p=0.24).

Because this data suggest "older" parents are able to rate more accurately, we wanted to take a deeper look at why this might be true. We discovered that worker age was a by-product, in most cases, of how long a worker had been a parent. Workers who had been parents for 16 or more years (n = 5) were more accurate in their ratings than

those who had been parents for 5 years or less (70.7% vs. 53.6%, respectively; Fisher's exact p=0.006).

These results suggest there is a subset of the crowd with substantially more expertise in predicting how parents of young children would judge the sentimental quality of audio recordings. This specialized crowd, as it were, appears to consist of middle-aged (and older) workers who are parents, and if they have been parents for a significant amount of time (16+ years), they seem to have even more expertise. If this is true, it would imply that their lived experience as parents has equipped them with specialized knowledge they are drawing upon for this problem. It is important to note that all the parents in the user study have been parenting for nine years or less. This could mean that a specialized crowd for curating digital audio recordings might need to have been parenting at least as long as the end users in order to be most helpful.

4.6.4 Kurator as a tiered architecture is effective

The goal of Kurator's tiered architecture is to allow for contributor types with different strengths to be traded off. For example, we use machine learning as a scalable, cost-effective way to take a quick pass, but human insight (i.e., from the crowd) is required to make more accurate judgments.

Measuring the performance of our approach requires looking at the ability to trade off cost and accuracy. Table 5 shows the tradeoffs induced by the crowd's performance as well as the tuning of the machine learning system used in our trials. Crowds are able to more accurately assess memories, but can be cost prohibitive. In fact, for 10,000 audio clips (about 8 months of data for our average family) it would cost $2,100 to have the crowd rate them all. Adding in the machine learning tier as a pre-filter for what the crowd

sees can reduce the cost by $711 if recall of Definitely's is optimized for ("Album") or by $2,076 if precision of Definitely's is optimized for ("Best Of"). This is exactly the intended effect.

Since the machine learning component itself can be tuned to trade off precision and recall (along a ROC curve that is specific to the classifier), it is possible for users to adjust the effect of the classifier to fit their preferences. This feature was not implemented in this prototype, and is left for future work.

Table 5. Curation quality, reduction in user effort, and cost savings caused by the machine learning tier of Kurator. "Album" favors recall of Definitely's, and "Best Of" favors precision for Definitely's (quality = precision for Definitely's, and %reduction = proportion of collection rated as NoWay)

	ML	
Regime	Album	Best Of
quality	76%	43%
%reduction	34%	99%
savings-600	$ 43	$ 125
savings-10k	$ 711	$2,076

4.6.5 Re-training the crowd

We believed that using feedback from the family would improve the quality of the ML and crowd's performance. To check this, we obtained family-specific guidance to the crowd for two families. This guidance came from responses to interview questions where parents were asked what they would tell a stranger in order to help them rate the family's recordings. An example from Family A, who provided guidance for their Definitely preferences, was: *"Choose 'Definitely' if it makes you laugh or if it gives you an emotional response."* Another example from Family B, who had two Definitely, one Maybe, and one NoWay guidance statements, was: *"Choose 'Maybe' if the kids are saying something but you can't make out what they are saying."*

We used the guidance statements to update the crowd's task descriptions for each family and then obtained new ratings for all 120 recordings for each of the two families. For both families, the crowd's F1 score for Definitely's increased (.46 to .57, and .66 to .78) but decreased slightly for NoWay's (.80 to .73, and .82 to .75). For both families, the guidance to the crowd included specific criteria for when to rate a recording as "Definitely" keep, which seemed to cause the crowd to assign more Definitely ratings for each family than they did without this guidance. This increase in Definitely's led to more of the Definitely's being identified, and it also led to fewer NoWay ratings. This caused the Definitely recall to increase and the NoWay recall to decrease.

4.6.6 Re-training the ML from the crowd and family

We re-trained the ML, per family, using the crowd's ratings, and then using the family's rating. After re-training on the crowd's ratings, the ML had a slight improvement in F1 score (0.54 to 0.57) when predicting NoWay ratings, and it improved significantly in precision (0.43 to 0.67) when predicting Definitely ratings. This suggests the crowd's inputs primarily benefited the ML's assessments of Definitely-rated recordings, which is not unexpected, considering the crowd is more effective in identifying Definitely's than the ML is. After re-training the ML on the family's rating, the ML scored much higher in F1 (0.51 to 0.66) for NoWay ratings, meaning the ML had a better grasp of what to filter out of the collections after incorporating families' inputs. The larger improvement was in the ML's precision for Definitely ratings, where it increased from 0.43 to 0.69. Again, this is as expected, and it validates that re-training the ML using the crowd's and/or the family's examples benefits the classifier's effectiveness, particularly when predicting Definitely ratings.

4.6.7 What happened?

To understand more about where Kurator differed from the preferences of the parents, we compared Kurator's selection of Favorites, the top *k* on each family's Definitely list. We used this to understand what criteria parents were using and how they differed from what Kurator could determine.

Overall, as indicated before, parents were generally satisfied with Kurator's selection. One parent stated she preferred Kurator's list over having to listen to 120 recordings. Another parent was pleased that Kurator caught an important clip she had overlooked in her ratings. As mentioned, Kurator found children saying cute sayings such as the Princess Bride quote mentioned above. For that clip, the crowd rated it as a Definitely, and one worker remarked: "*It's really cute but dark! [It] would make a parent laugh*."

Other examples, such as a 2-second recording of a parent's two daughters laughing and making unintelligible, silly sounds, suggest the crowd was able to find content likely to be meaningful to parents even without much linguistic content.

Kurator also missed some clips. Some cases in which responses were counted as incorrect did not have an impact on the end users. For example, when duplicate clips (multiple recordings with the same content) were present, Kurator sometimes included all of them in its top *k*, or it would pick a different one than the family chose as a Favorite. This artificially decreases Kurator's measured performance.

One parent had two clips in her collection of her daughter saying "My name is Clementine." Although they sounded almost exactly the same, in one of the recordings, the parent heard her daughter use her "home voice", and thus selected that clip as the

Favorite out of the two, although she would have been happy with either: *"One example is Clementine had two and they were basically exactly the same, but I picked one because it sounded more like what she sounds like at home. She's very shy and she doesn't talk a lot to other people, so only really us and our family know what she really sounds like."*

Similarly, another parent had four clips of her son saying the same thing. She marked them all as "Definitely" in her first pass through the collection. However, when she had the chance to review her selections and upgrade some of them to Favorites, she only marked one as a Favorite: "*I think I had saved Cooper saying I love you 4 times, but then I [de-selected] 3 of them. I don't need him saying it 4 times."* Kurator however classified three of the four "*I love you's*" as top picks. She was not upset about this near-miss, because, although she only saved one to avoid duplication, these were some of the most valuable clips that she had, mentioning that *"... there wasn't any other better choice to choose from."*

This ability to pick up on meaningful content was a key strength of the crowd. Whereas automated curation strategies depend on surface-level features, and parents had a certainty drawn from their in-depth knowledge of their children and their own preferences, the crowd nonetheless was able to draw on its own experience to guess what might be meaningful quite accurately. We return to this idea of "common understanding" in the discussion section.

At times, Kurator severely missed. Often it was because of very specialized and idiosyncratic knowledge that only the family possesses. We believed this would be true, although future iterations of Kurator need to take them into account.

The one parent who expressed dissatisfaction with Kurator's list was disappointed that the list she heard included recordings mostly of only one of her three children (anonymized): "*I didn't hear a lot of Sally or Michael. Yeah, it was mainly Max. I mean that's just one from my child, I need to hear Sally and Michael. I'm an equal rights mom. All my children get to have one each.*" This comment potentially reflected curation preferences that favored representativeness in what was kept.

Where the ML and the family differed, the clips were unremarkable at the signal level: they were mono-tone or quietly spoken utterances. For 3 of the 4 clips where the crowd and the parents differed, workers had trouble understanding the children's poor articulation of otherwise normal English words. The fourth crowd-filtered Favorite was a whistle being blown, which the crowd deemed as "just noise". The parent explained: "*That was when we went for Ella's birthday and they all got those Chuck-E-Cheese whistles. [The kids blew the whistles] in the car, the whole way home and the whole next day. Of course, [KidKeeper] got some Chuck-E-Cheese whistles, it was hilarious! ... I had already forgotten about it and that reminded me how awesome it was."*

The sound of the whistle was a trigger of a particularly sonic memory for this parent, but would be perceived by anyone else as just noise. The parent anticipated the obscure nature of the clip, and did not expect the system to catch it because "it was an inside joke." In these examples, "insider" knowledge is required, and Kurator failed expectedly. This suggests that the ML and the crowd may need to be tuned conservatively for some families.

4.7 Discussion

Parents were generally happy with Kurator's top *k* even though the quantitative data paints a bleaker picture.

The findings suggest that crowds are effective at predicting whether specific children's digital audio recordings will be valuable family memorabilia. As mentioned, leveraging crowds, alone, is not a scalable solution. Combining mixed expertise crowds with machine learning increases scalability and decreases monetary cost, but it comes at a price in terms of system precision. Kurator's ML component was fairly effective, at best, but the paid crowd was very effective. The combination of the two resulted in a precision that was less than what the crowd could achieve alone. For the tiered architecture, researchers must investigate more deeply what tradeoffs their users want, in terms of price versus precision.

4.7.1 Leveraging the crowd's common understanding

Curating subjective, semantic content has been theorized to be beyond the current capabilities of automated approaches. Barriers range from inability to make idiosyncratic judgments to a lack of contextual knowledge needed. Further, the criteria that those with "expert" knowledge have are difficult to fully articulate. Yet, with Kurator, the combination of crowd and machine was somehow able to be reasonably successful.

There were obvious wins. As we point out in our findings, some clips were selected to keep unanimously by the system and the family. These clips were those that were commonly understood to be good and meaningful to a parent. These common understandings, for example recognizing that a child speaking about a parent was highly likely to be valuable or that "I love you" is always worth saving, were critically important

to the success of Kurator. We found that the crowd was underestimated in the literature with regard to its ability to react sympathetically to a subjective task. Their ability to pull from common cultural assumptions (Berger and Luckmann, 1991) is actually a viable source of help on this class of problems.

There were also cases where Kurator consistently failed. In these cases, the value of a clip could not be surmised from its content alone. For some clips, the sound quality of the recording, due to it being noisy or unclear, caused it to be filtered out by the crowd or ML. For other clips, such as an "inside joke", a common understanding of cultural assumptions was not helpful. In the third case, the value of a clip was relative to its role in a collection rather than just its own content. In each of these cases, Kurator did not have the necessary context and information to make a good guess. However for most of these cases, Kurator was expected to fail. It was taken for granted by parents that there were some audio clips that would be impossible to recognize to anyone but them.

There were hard cases, however, where the value of a clip was more ambiguous. In these cases, the crowd's common understanding was not nuanced enough to recognize the full value of a clip. Yet, the crowd was not consistently wrong. For cases where the crowd missed the nuance, such as failing to recognize a child's message to her father to not leave for work, there were corresponding cases where the crowd actively recognized non-obvious semantic value and even crafted elaborate narratives to explain why they thought it might be valuable to a parent. The ability to recognize the value of some clips may be tied to workers' experience or ability to create a believable narrative for themselves about the potential value of the clip.

4.7.2 Criteria may change over time

For some parents, their ratings on the first few recordings were not as consistent with the rest of their ratings due to a lack of familiarity with their digital collection. This lack of familiarity impacted their curation decision-making criteria, and it potentially created noise in the system, which could mislead machine learning agents. One possible remedy for this is to prevent users from rating the first time they are presented with a refined collection from a curation system. Another solution might be to have parents re-rate old recordings, either randomly-selected or strategically-selected ones. We elaborate on this below.

Some parents speculated that their criteria would change over a longer period of time, like 20 years. Prior research has shown that time impacts the meaningfulness of certain digital content (Gulotta et al., 2015). Ceroni et al. (2015a) also point out their participants desired to be able to update their preservation decisions every 2-5 years. The implication is that we need a way to encourage long-term use of a curation system. We know parents enjoyed the task, especially in small time chunks, which makes an approach like selfsourcing look promising (Teevan et al, 2014).

Curation systems like Kurator could incorporate a selfsourcing framework as a way to re-calibrate the crowd and machine learners as the family's notion of what makes an item meaningful changes over time (Gulotta et al., 2015).

4.7.3 Focus on specialized crowds

Using specialized crowds is another avenue for potential improvement. Because Kurator leverages a mix of crowds with varying expertise levels, it may benefit the system if it were possible to identify and leverage specialized crowds dynamically. If a

subset of a paid crowd had specialized skills, particularly with how and what to curate for long-term preservation, Kurator could leverage their higher levels of expertise. At least some portions of the crowd seemed to use an in-depth thought process when making their decisions about ratings. We believe there is some amount of common understanding, such as looking for "cuteness", but the stories the crowd members were telling indicate they were going past "cuteness", per se.

An investigation into specialized crowds would need to, first, identify them and, second, determine how to dynamically leverage them. This study has shown it may be possible to identify specialized crowds within the paid crowd. Dynamically using them may be more difficult, but specialized tests, or discount expertise metrics, may be helpful in automatically identifying, then utilizing, those with expertise in curating. In the next chapter, we more fully explore ways to identify and leverage specialized crowds.

4.7.4 Privacy

One significant obstacle to deploying crowd-powered content curation systems is privacy. To help partially address this issue, we used a less-identifiable medium (audio), kept user information private, used large distributed crowds, and randomly ordered content to prevent workers from “following” certain users or families. However, families may share (accidentally or intentionally) content that contain sensitive data, personal information, or other private content. Prior work has shown that there are several means by which workers can access or even reconstruct shared sensitive user information (Lasecki et al., 2014). Obfuscation methods such as audio warping and worker routing that minimizes the amount of information from one family that one workers sees can further improve the chances of safe use of crowd powered systems in our setting.

Research has also explored how intelligent division of content (Kajino et al., 2014; Lasecki et al., 2015) can help reduce the threat of information exposure. Future work will explore how crowd's ability to assess the sentimental value of content is affected by these filters.

4.8 Limitations

Our study explored the viability of using crowds and our tiered architecture for a curation task with a focused group of participants. A larger scale deployment over a longer period of time is needed to further explore questions about how people choose to trade off quality and cost, how assessment of sentiment changes over time, and how much the curation regimes between families diverge (or possibly even converge) over time.

Another limitation of our study is that we only collected rating data from one parent in each family. A future direction for this work could be collecting ratings from more than one family member, and perhaps even close friends of the family, to allow for a deeper investigation into the variance of ratings within the "expert" crowd, and between the expert crowd and a specialized paid crowd.

Finally, Kurator used a majority voting mechanism to determine the crowd's rating on a particular audio recording. We believe this could be less efficient than weighted voting. A promising future direction for Kurator, particularly in the context of a longitudinal study, would be to track crowd workers' rating accuracies over time and then use a single, weighted rating on a recording when a known worker is involved. This would alleviate the need to obtain multiple ratings for every recording, which is yet

another way to leverage mixed expertise in the paid crowd to benefit system performance.

4.9 Conclusion

In this chapter, we have introduced Kurator, a proof of concept for hybrid intelligence systems that use mixed expertise between crowds in a tiered architecture to synthesize inputs from multiple layers of contributors, such as machine learning, the crowd, and the family. We applied Kurator to the problem of reducing the burden of curating a family's digital audio memories. Our results demonstrate that paid crowds can accurately select content that parents find sentimental even without specialized context, and that machine learning can effectively be used to trade off accuracy versus cost. Over time, families, who are ostensibly the experts for this task, can contribute directly to their own curation tasks as well as provide feedback to paid crowds and the machine learning components of the system to get more accurate, personalized results.

The problem of personal digital media curation falls within the second problem called out by this thesis, which are personalized, subjective problems with multiple valid solutions to an end user. We have shown that leveraging mixed expertise in this problem area is worthwhile, and we have answered the second research question posed in this thesis:

- **RQ2.** Under what conditions, and to what extent, is there benefit when using a mix of crowds, differentiated by types and levels of expertise, to solve a problem when the crowds are working on similar tasks?

As well, Kurator represents a workable solution that uses opinion-based expertise criteria. Using Kurator, we were able to identify specialized crowds, but we wanted to

investigate more fully how we might identify and leverage specialized crowds, particularly subcrowds within a crowd, differentiated by expertise. Also, we wanted to better understand how a system might work in the middle of the fact-opinion spectrum of expertise criteria. The next chapter explains a study of the Question Finding problem and how it achieves this, which answers our third research question.

Chapter 5. Question Finding: Mixed Subcrowds and Expertise

In this chapter, we prototype an approach that builds on what we found with Escalier (mixed expertise within a crowd) and Kurator (mixed expertise between crowds) by leveraging multiple subcrowds, differentiated by expertise. Our study of this approach is a culminating study for this thesis, and it aims to answer the third research question:

- **RQ3**. Under what conditions, and to what extent, is there benefit when using a mix of subcrowds within a crowd, differentiated by expertise?

Answering this question will show that there is benefit to using subcrowds, which are groups from the same crowd but differentiated by their topic-specific expertise. We refer to these subcrowds as specialized crowds.

In the previous chapter, Kurator shed light on the importance of specialized crowds, but we wanted to investigate more fully how we might identify and leverage specialized crowds. We do this by studying a problem we call "Question Finding".

The goal of Question Finding (QF) is somewhat backward to the standard Q&A site. Instead of having people answer one another's questions, we wish to create questions where the sophistication of an answer, based on pre-existing material, will align with the sophistication of the question. QF overlaps both problem areas in the scope of this thesis: problems where the diverse expertise of a population must be leveraged to uncover more of a solution space, and personalized, subjective problems where there are

multiple valid solutions to an end user. To do address this problem, we use a combination of different types of expertise from the fact-opinion spectrum (see Chapter 2): fact-based criteria to distinguish who, in the crowd, knows what, and opinion-based criteria to add human subjectivity into the measurement process.

5.1 Introduction and Background

Question Finding (QF) is a general problem for information sites that lack a convenient way to direct a user, who has an information need, to suitable content containing the answer. We studied this problem using the NASA website[4] and its myriad sub-sites[5] as a motivating example. NASA has a number of websites that are poorly designed for lay people to conveniently find suitable answers to their questions. Not only are there many websites covering a large number of topics, even within a single topic, there is a wide variance in the sophistication of the content. Some content may be too advanced to be a suitable answer to a beginner-level information need. As well, some content is too basic to be a suitable answer to an expert-level information need.

To ground the problem of QF to something familiar, consider this scenario: NASA has much web content about stars, and we would like a range of questions from an elementary school-level, such as "why is the sun so bright?" to more complex questions, such as "What physical process causes the sun to be so bright?" Then a user could be directed to elementary-level articles where he might learn that "it's because it is a star that we are the closest to" or to more advanced content on astrophysics, where he would learn about "the thermonuclear fusion of hydrogen into helium."

---

[4] http://www.nasa.gov/
[5] http://www.nasa.gov/sitemap/sitemap_nasa.html#.V6FcP9ArJE4

In this chapter, we prototyped an approach for generating questions of varying complexity for multiple topics. This approach identified specialized subcrowds of workers, differentiated by scientific topic and expertise about that topic, and leveraged them to generate a variety of questions. Because people's self-reports are unreliable (Donaldson and Grant-Vallone, 2002 and Arnold and Feldman, 1981), we used a survey instrument, distributed via crowdsourcing, to obtain expertise measurements as well as to solicit questions from the crowd. The expertise measurements are used to identify topics and levels of expertise for crowd workers. Knowing workers' expertise levels allowed us to examine if expertise is correlated with the complexity of questions they generate. We found that a range of expertise in a topic is needed to reliably generate questions that vary in level of difficulty. As well, we determined the crowd consists of multiple specialized crowds who show a range of expertise for multiple topics. Therefore, this approach pulled mixed expertise and specialized crowds together to create questions differentiated by levels of complexity for multiple topics.

The remainder of this chapter is structured as follows: we first explain the goals and design of our prototype, then we describe how we evaluate it, then we explain the findings from the evaluation, and then we discuss the salient lessons learned, the limitations of this study, and our conclusions.

5.2 Prototype Design

In this chapter, we use the term "prototype" to refer to an approach, not necessarily a technical prototype. We prototyped an approach with two goals in mind. The first goal was to identify specialized subcrowds, each having a range of expertise levels, differentiated by topical expertise. To do this, we needed measures of expertise

for multiple topics. Just as important, we also wanted to know when we could not identify expertise levels and specialized crowds.

The second goal was to solicit questions from the crowd that can be differentiated by complexity, and to determine if these differences are correlated to workers' expertise. We also wanted to know what difficulties people had in doing this.

There were three important considerations we had to work through. First, we needed an instrument to measure workers' expertise levels in various scientific topics. Second, we needed an approach for question solicitation that would encourage crowd workers to create questions with rich content and of varying complexity. Third, we needed to determine a way to measure the difficulty level of the questions the crowd generated. Next, we discuss each of those considerations, in turn.

5.2.1 Using a Validated Measurement Instrument

For the measurement instrument, we leveraged prior work in measuring civic scientific literacy (e.g., Miller 1998, 2012a, 2012b). We used a validated survey instrument, with suitably updated questions, for measuring crowd workers' expertise levels. The survey consisted of 42 questions: 35 closed-ended questions (i.e., multiple choice and true/false) and 7 open-ended questions (i.e., written responses). The questions fell into two broad domains: biology and space. Within the biology domain, there were questions covering the topics of DNA, stem cells, molecules, evolution, and genetics. Within the space domain, there were questions covering the topics of planets, the Sun, Earth's atmosphere and geology, and the universe.

The percentage of correct items for a topic is used as the measure of expertise in that topic. To determine this measure, all closed-ended responses were weighed on the

same binary scale (i.e., one point if correct, zero if not), and the open-ended responses were weighted using a three-point scale, effectively making it worth three times more than a closed-ended item. This weighting of the open-ended responses accomplished two things: it allowed for a more fine-grained measurement of varying quality, and this higher resolution allowed us to tease apart the workers on the high end of the expertise scale. This treatment of open-ended responses is similar to Miller's measurement approach for open-ended responses (1998, 2012a, 2012b).

We used this measurement approach to establish measures for well-represented topics in the survey. In particular, we established measurements for the space and biology domains, and as a subset of those domains, we had measures for the topics of DNA, stem cells, the solar system, and the universe.

5.2.2 Soliciting Questions from the Crowd

We needed an approach for question solicitation that would encourage crowd workers to create questions with rich content and of varying complexity. To build our intuition on how to design for this, we performed several pre-studies. We found that the scope of the topics might affect the crowd's ability to create questions that varied in complexity. This is a classic problem in expertise finding on where to divide the topics and how narrow should they be (Merritt et al. 2016). Topics that were too broad resulted in vague questions, and topics that were too narrow or specialized (e.g., molecular biology), caused people to struggle with having enough knowledge to generate multiple questions. Ultimately, we stayed within the science and technology domain, and pre-testing pointed us towards four topics that showed promise: DNA, stem cells, the solar system, and the universe. As mentioned, the data collection instrument can be used to

measure expertise in these topics, which allowed us to correlate workers' topical expertise with the difficulty level of their questions.

The pre-studies also showed us that soliciting multiple questions per topic led to richer questions because it forced workers to go beyond the canned "What is *[insert topic here]?"* This also applied to asking the crowd for multiple levels of difficulty: instead of simply asking for questions, we asked for easy, medium, and hard questions. Not only did this increase the number of questions we asked for, it also prompted intentional thinking about the difficulty levels of questions, which we found the crowd was able to do.

5.2.3 Measuring Question Difficulty with a Rubric

To measure the difficulty level of the crowd-generated questions, we created a rubric for assessing the complexity of the question. The rubric was also used to score the open-ended responses in the survey. In short, we considered the complexity in the question content as well as the complexity of the answer. If the question or the expected answer covered a complex topic, the scoring would reflect that. Although the rubric adds consistency and reliability to the scoring, the scoring will always include some amount of subjectivity. Thus the use of the rubric to score the questions generated by the crowd, as well as the open-ended responses in the survey, introduced opinion-based criteria into the measurements of expertise. Combined with the fact-based measurements from the closed-ended responses on the survey, this study is positioned somewhere in the middle of the fact-opinion spectrum of types of expertise.

5.3 Evaluation

As mentioned, the prototype had two goals: identifying specialized subcrowds by topical expertise, and soliciting questions with varying complexity. To understand how the prototype met these goals, we evaluated it using Amazon Mechanical Turk as the crowdsourcing platform, limiting participants to workers located in the U.S.[7]

To measure success for the first goal, we used a combination of analyses:

1) a visual analysis of the distribution of worker scores for each measure,
2) a paired t test between measures,
3) and the Pearson correlation coefficient between measures.

For (1), the distribution of scores should show that a range of scores exists. If workers all achieve the same score (i.e., they cannot be differentiated because the t-tests are not statistically significant), then either the measure is invalid, or the crowd is homogenous in its knowledge level for that subject matter. Either way, we cannot conclude that the measure is able to identify a range of expertise in the crowd.

For (2) and (3), we considered these results together. In short, (2) tells us if workers score differently on the two measurements being tested, and (3) tells us if a worker's score on one measure is correlated with their score on the other measure. For example, if workers' scores on both measures are found to be the same, on average, according to (2), and the scores on both measures are perfectly positively correlated, then we can conclude these two measures are either measuring the same thing, or a worker's level of knowledge in one subject perfectly predicts their level of knowledge in the other.

---

[7] The validated survey instrument we used was validated on U.S. participants.

Either way, we cannot conclude we have two *distinct* measures because they are redundant.

To measure success for the second goal, we compared the questions generated by the crowd to determine two things:

1) if a range of complexity exists in the questions,

2) and if the complexity of the questions aligns with workers' expertise levels.

For (1), we scored the easy, medium, and hard questions created by the crowd, and then we compare these scores. In these comparisons, we were looking for a significant difference in the average complexity of questions in the easy, medium, and hard categories. If there were differences, then we concluded that the crowd was able to generate questions with a range of complexity levels. For (2), if higher expertise workers ask harder questions, and lower expertise workers ask easier questions, then there is support in concluding that workers' expertise levels are general predictors of their ability to create complex questions.

Using these measures of success in evaluating the prototype, we discuss the findings next.

5.4 Findings

We collected data from 120 Mechanical Turk workers, which consisted of responses to the survey and six questions generated for each of the four topics (DNA, stem cell, the solar system, and the universe). The six questions consisted of two easy, two medium, and two hard questions.

In this section, we discuss the findings from the evaluation of the prototyped approach's two goals. We also discuss the challenges faced by the workers, as reported by them in open-ended feedback sections on the survey.

5.4.1 Goal 1: Identifying specialized subcrowds, differentiated by expertise

The first prototype goal was to identify specialized subcrowds, each having a range of expertise levels, differentiated by topical expertise. To accomplish this, we scored the workers' responses on the data collection instrument, which produced measurements of expertise for the space and biology domains, as well as the sub-topics of DNA, stem cells, the solar system, and the universe. We thus needed to determine if these measurements were valid, according to the three-step analysis discussed previously. First, we discuss the domains of biology and space, and then we discuss the narrower topics of stem cells, DNA, solar system, and universe.

First, we wanted to determine if the biology and the space expertise measures were valid. The first step is to analyze the distributions of biology and space scores. In Figure 9, we can see that both measures have scores that span the range from 0.3 to 1.0 (note the scale is 0.0 to 1.0), and there are many workers represented in almost all segments of the histogram. This data indicates we can use the biology and space measurements of expertise to identify a range of expertise levels in those domains. In contrast, if all workers scored the same or nearly the same in a domain, then it would mean the prototype was not able to identify a range of expertise levels within that domain.

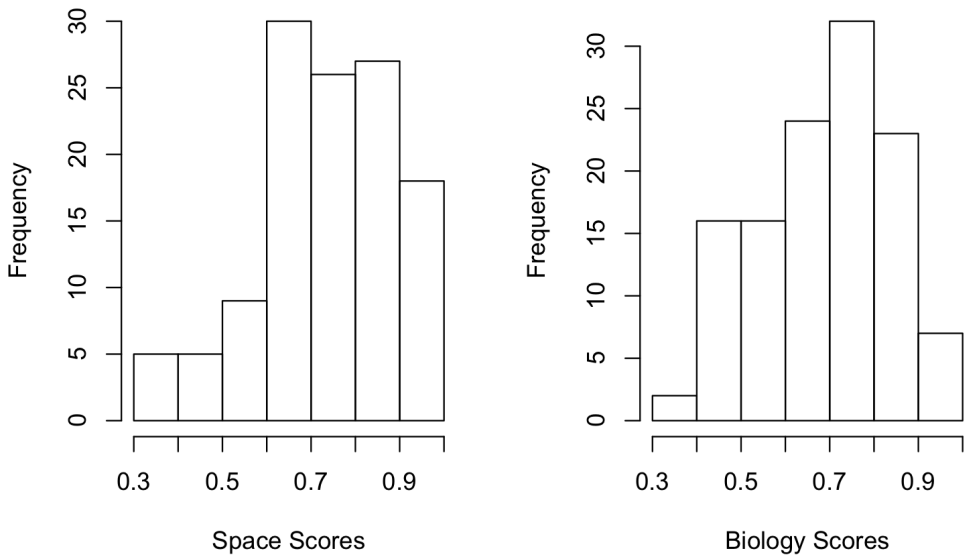
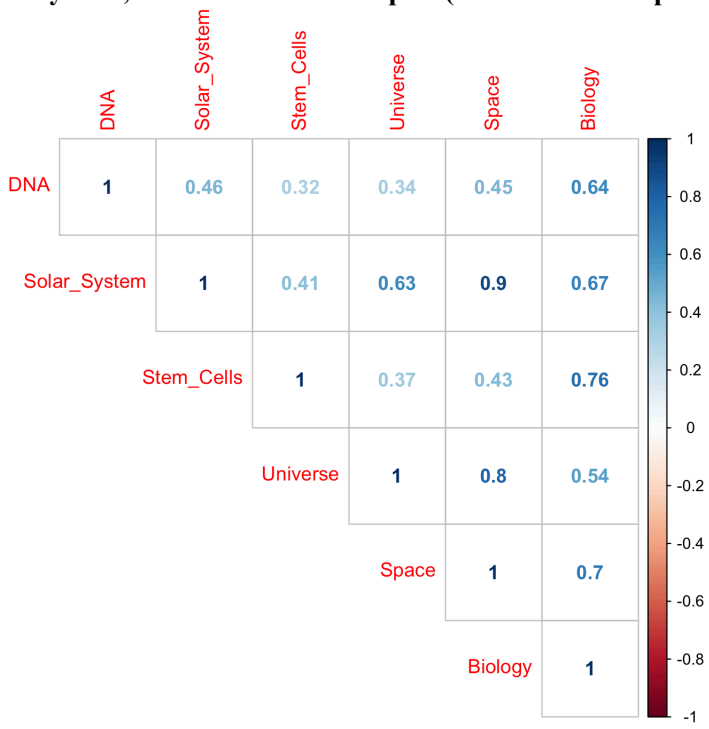


Figure 9. Histogram of Biology and Space Expertise Scores for 120 crowd workers. 0.0 to 1.0 is the range of the scale that each expertise measure uses.

The next step is to determine if the space and biology measures are distinct from each other, which would indicate whether they are measuring expertise in *different* domains. Thus we analyzed the difference between workers' space and biology scores. For this, we performed a paired t test, which measures the mean of the differences in scores, per worker. This test revealed a mean difference of 0.04 that was statistically significant ($p<0.001$). As well, there is only a moderate correlation in workers' scores for each domain (see **Error! Reference source not found.**), with a Pearson correlation of 0.70. This suggests that a worker's score in the space domain is different than their score in the biology domain, but there may be a slight trend where a higher score in one domain coincides with a higher score in another domain. These results indicate the space and biology measures of expertise can be used to identify specialized crowds in those domains.

Table 6. Correlation Matrix for Biology and Space domains as well as DNA, Stem Cells, Solar System, and Universe sub-topics (Pearson's *r* is reported).



Second, to determine if the prototype was able to identify even narrower topics, we followed this same three-step analysis pattern for the four subtopics: DNA, stem cells, solar system, and universe. Figure 10 shows the histograms of scores for each of these measures. We can see that all measures have scores that span the range from 0.0 to 1.0, and there are many workers represented in most segments of the histograms. Note that there are fewer bins in these histograms than the ones in Figure 9 because there are fewer items with which to measure, which means the number of possible scores is smaller. This data indicates the prototype is able to identify a range of expertise levels in each of these four topics.

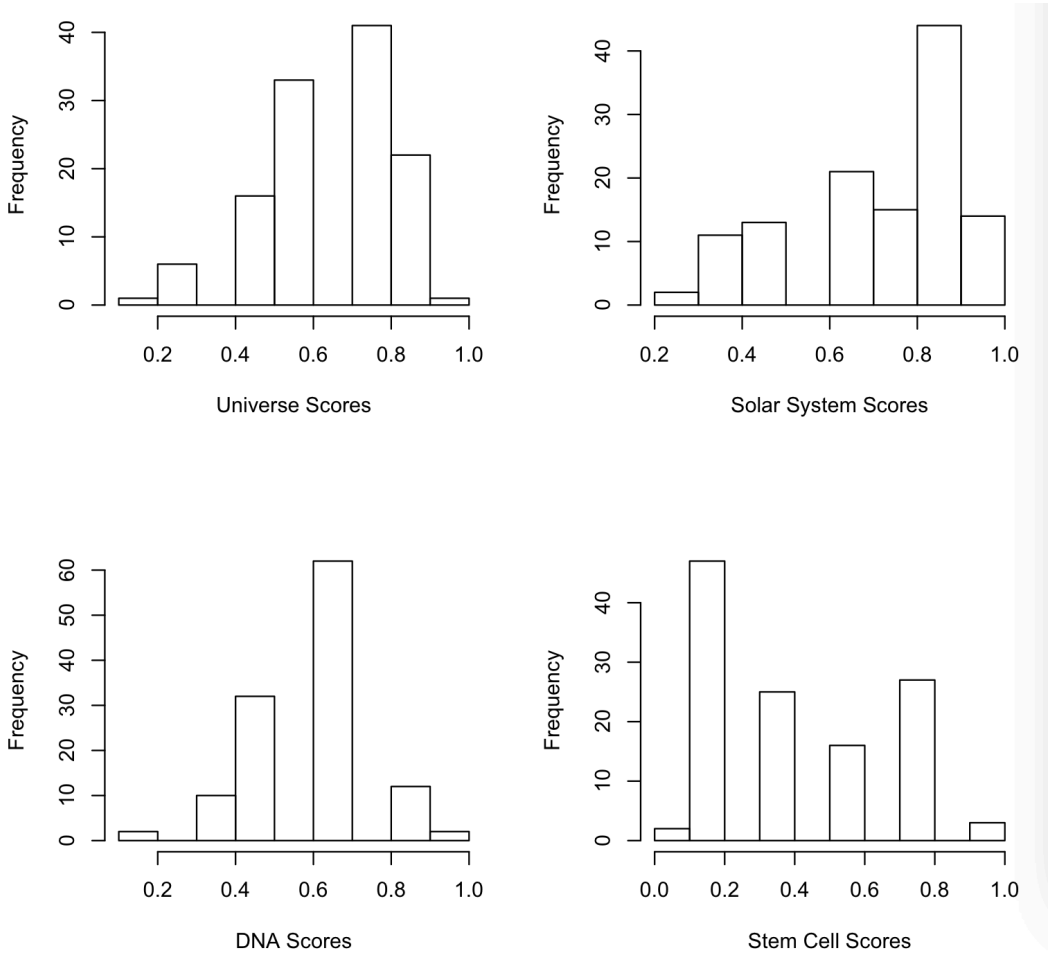


Figure 10. Histograms of Universe, Solar System, DNA, and Stem Cell Expertise Scores for 120 crowd workers. 0.0 to 1.0 is the range of the scale that each expertise measure uses.

In Table 6, note that all correlation coefficients are positive, indicating a general trend where workers who score higher on one measure tend to score relatively higher on other measures as well. This may be due to these topics being widely taught as part of general science education in the U.S.[9] (Miller, 2010), but additional tests are needed to say this conclusively. To finish the three steps of analysis for these measures, we analyze the biology domain and topics, then the space domain and topics, and then all the topics.

[9] We remind the reader that the survey was only available to Mechanical Turk workers located in the United States.

For the biology domain, the topical measures for DNA and stem cells are distinct from each other and the biology measure. These differences were statistically significant ($p<0.01$). Also, the correlation between DNA and stem cells (0.32) is weak, and the correlation between DNA and biology (0.64) is weak to moderate. The stem cell scores are moderately correlated with the biology scores (0.76). This suggests that workers with higher expertise in stem cells are likely to have higher expertise in biology, which makes sense. These results indicate that we have three distinct biology-related measures of expertise--an overall biology measure, and stem cell and DNA measures--that can be used to identify multiple specialized crowds.

For the space domain, the measures were more strongly correlated than in the biology domain. We found a moderate correlation (0.80) between the universe and the space measurement instruments, but the space score was 0.09 points higher than the universe score, on average, and it was statistically significant ($p<0.001$). This indicates these two measures identify separate specialized crowds, but workers scoring high in space likely score high in universe as well.

The space and solar system measures, however, were strongly correlated (0.90) with a mean difference of 0.005 points ($p=0.5$). This fits our notional example (see Evaluation section) where we cannot conclude these are two *distinct* measures because they appear redundant. There are a couple possible reasons for this outcome. One possibility is that workers' expertise level in one topic is very similar to their expertise level in the other. Another possibility is that the solar system questions in the survey were not specific enough, thus making it another measure of general space expertise. Thus we cannot say conclusively that the solar system measure is a standalone topical

expertise measure. The universe and space measures, however, are distinct expertise measures.

We also tested the DNA and stem cell measures against the universe measure. All correlations were weak (<0.40), indicating that higher expertise in one topic does not necessarily correlate with higher expertise in the others. Although the stem cell and universe scores were different ($p<0.001$), the difference in the DNA and universe measures was not statistically significant ($p=0.08$). In Figure 11, we see that there are some workers scoring moderate to high in DNA but low in universe, and other workers score moderate to high in universe and low in DNA. Thus we conclude that the DNA, stem cell, and universe measures of expertise are distinct, and they can be used to identify specialized crowds in each of those topic areas.

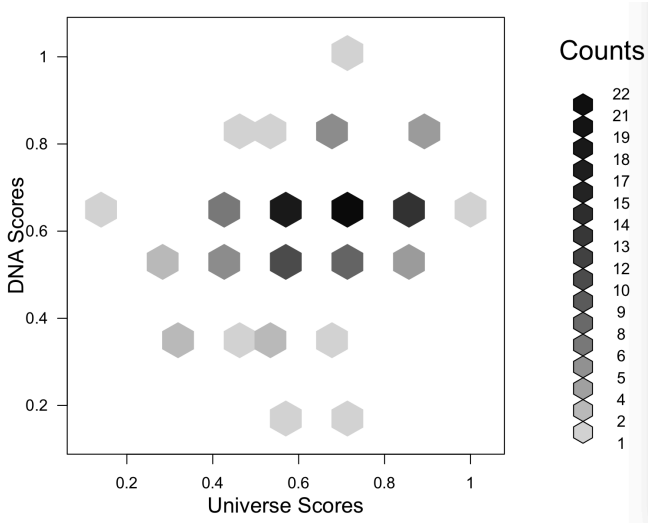


Figure 11. Scatterplot of DNA scores versus Universe scores.

In summary, the first prototype goal was to identify specialized subcrowds, each having a range of expertise levels, differentiated by topical expertise. The results in this section indicate the prototype is able to use expertise measures for the biology and space domains, as well as the DNA, stem cell, and universe topics, to identify specialized

subcrowds in those domains and topics. We also found that the prototype's use of the solar system measure of expertise was redundant with the overall space measure, meaning we cannot say conclusively that we have a valid measure for solar system expertise.

5.4.2 Goal 2: Differentiating crowd-generated questions by difficulty level

The second prototype goal was to solicit questions from the crowd that can be differentiated by complexity, and to determine if these differences are correlated to workers' expertise. As mentioned, we solicited two easy, two medium, and two hard questions from each crowd worker for each of four topics: DNA, stem cell, the solar system, and the universe. We scored the DNA and universe crowd-generated questions but not the stem cell and solar system questions. After a brief discussion of what the stem cell and solar system questions taught us about the proper scoping of topics, we then move into a discussion of how the crowd was able to generate DNA and universe questions with suitable levels of difficulty.

5.4.2.1 Workers lacked depth of knowledge for some topics

A review of the stem cell questions gave us insight into issues with the subject matter itself. Apparently, the topic of stem cells was not a familiar one to the workers, and most seemed to struggle with the concept. As a point of reference, the average score on the stem cell expertise measure (0.45) was well below all the other measures (next lowest is 0.61). Over half of the workers received a zero score for their open-ended response to explain what a stem cell was. This lack of knowledge caused workers to create irrelevant questions. The questions that were relevant were often overly vague and a variation on the same few types of basic questions, such as: "what are stem cells?",

"what do they do?", "where do they come from?", "why are they controversial?", and "can stem cells cure [insert illness here]?".

In the open-ended feedback for the stem cell questions, all the top tier workers said they had difficulty with these questions. One said it was difficult coming up with easy questions for stem cells, and two others said they did not "know enough" or have "advanced knowledge" to ask hard questions. These results indicate there may be some topics that are too complex for there to be a diversity of expertise levels amongst the general public[12]. Thus measuring a diversity of expertise in complex topics is likely to be very difficult, if not impractical.

Similarly, the solar system questions lacked depth, but for different reasons. Most workers seemed quite familiar with the makeup of the solar system (planets, asteroids, moons, comets, the sun), as evidenced by workers scoring the highest, on average, on the solar system measure (0.74) as well as the one open-ended question pertaining to the solar system. Although these high scores seemed to indicate a widespread familiarity with the solar system, there was not much depth in the questions. A substantial majority of the questions were simple questions of fact, like "is x bigger than y?", "how far is x?", "which planet has the biggest/most...?", "how big is...?", or "how many...?".

It is possible that the topic of "solar systems", while familiar to many, lacks depth as a standalone topic. Attempts to go beyond surface-level knowledge of the topic immediately leads into much more advanced subjects, such as planetary science or astrophysics. This notion of lacking depth was summarized by one astute worker, who

---

[12] We are not claiming Mechanical Turk workers are an accurate representation of the general public, but we do view the workers as having access to *at least* as much information as the general public, as evidenced by their participation in an online work environment.

made the comment: "It seems most of the questions involving solar systems are relatively easy, but more complex questions involve theory as to what happens during formation and eventual system collapse." Again, these results suggest there may be some topics that are not too complex but, instead, are too shallow for there to be a depth of expertise without transitioning into other (possibly more complex) topics. This problem of topic scope was further elaborated on by another worker, who stated: "Solar system is a very generic subject, so it sort of made it easy to come up with questions, but also difficult in that it could refer to many different things." We discuss topic scoping in more detail later in this chapter.

5.4.2.2 Crowd was able to generate questions that varied in complexity

Although some topics were not conducive to creating question with varied complexities, we found that the crowd was able to generate DNA and universe questions that varied in complexity. In this section, we explain the findings that supported this conclusion, and we do it in combination with our analysis of whether workers' expertise levels affected their ability to generate easy, medium, and hard questions.

To determine the difficulty of questions, we scored the questions created by workers from the top 10%, middle 10%, and bottom 10% based on the universe and DNA expertise measures. As mentioned, we used a rubric to score these difficulty levels. Scores were on a three-point scale, where one meant a question was easy, two meant it was medium, and three meant it was hard. For both topics, if a question was irrelevant, it did not receive a score. We had initially given irrelevant questions a score of zero, but this artificially reduced the mean scores.

This analysis is done in two parts: we first evaluate the universe questions, and then we evaluate the DNA questions.

First, in the universe topic (see), the bottom tier of workers was able to ask easier questions (mean = 1.05) than the middle and top tier could produce; this difference was statistically significant ($p<0.01$ for both). The bottom tier's medium and hard questions were the same difficulty level, which was more difficult than their easy ones ($p<0.01$). This indicates the bottom tier of workers might only be able to generate two classes of questions: easy and medium difficulty questions. Their easy questions are likely to be easier than what the middle and top tier workers are able to create.

Table 7. Question difficulty scores for Universe topic. Bottom, middle, and top 10% groupings are based on workers' scores on the universe expertise measure.

Universe Question Difficulty Scores				
Worker-reported Difficulty	*Worker Grouping*	*Mean Difficulty Score*	*n*	*std dev*
Easy	Bottom 10%	1.05	21	0.22
	Middle 10%	1.36	25	0.49
	Top 10%	1.35	23	0.49
Medium	Bottom 10%	1.45	22	0.51
	Middle 10%	1.48	23	0.67
	Top 10%	2.00	24	0.51
Hard	Bottom 10%	1.39	23	0.66
	Middle 10%	1.62	21	0.80
	Top 10%	2.17	24	0.56

With the middle tier workers, there was no statistically significant difference between the average scores for their easy, medium, and hard questions, and these scores were in the medium difficulty range. This suggests the middle expertise tier for universe can ask a variety of medium difficulty questions but may be inconsistent in creating easy or difficult ones.

The top tier of workers in the universe topic was able to ask the most difficult questions. Their medium questions were harder than the middle tier's hard questions, on

average, but this difference was not significant ($p=0.06$). This suggests the top tier is a reliable source of hard and medium/hard universe questions, and they, like the middle tier, are not able to generate easy questions consistently.

Second, in the DNA topic, the differentiation existed but was not as clean (see Table 8). There was no statistically significant difference between the expertise tiers for the easy and medium question difficulties. The easy questions were about the same difficulty regardless of expertise tier. The medium difficulty questions were also indistinguishable between expertise tiers, but they were noticeably more difficult than the easy questions ($p<0.05$). The middle and bottom expertise tier's medium and hard questions were basically an equivalence class of medium-hard questions. The questions in this equivalence class were more complex than the easy questions, with statistical significance.

Only the top expertise tier was able to differentiate easy, medium, and hard questions with statistical significance. As well, the top tier's hard questions were more difficult than the bottom tier's hard questions ($p<0.05$), but not more difficult than the middle tier's hard ones. As mentioned, the middle tier's hard questions were not significantly harder than the bottom tier's. This indicates the middle tier is not consistent in creating harder questions (1.85), on average, than the bottom tier (1.62), but the top tier is likely to be consistent in creating harder questions.

Table 8. Question difficulty scores for DNA topic. Bottom, middle, and top 10% groupings are based on workers' scores on the DNA expertise measure.

DNA Question Difficulty Scores				
Worker-reported Difficulty	*Worker Grouping*	*Mean Difficulty Score*	*n*	*std dev*
Easy	Bottom 10%	1.25	20	0.44
	Middle 10%	1.20	25	0.41
	Top 10%	1.29	24	0.46
Medium	Bottom 10%	1.62	21	0.67
	Middle 10%	1.69	26	0.49
	Top 10%	1.63	23	0.5
Hard	Bottom 10%	1.62	21	0.67
	Middle 10%	1.85	26	0.37
	Top 10%	2.04	24	0.69

As discussed, the crowd-generated DNA questions, when stratified according to workers' score on the DNA expertise measure, were not cleanly differentiated in many cases. To unpack this, we reviewed worker feedback on this task. Interestingly, a top-tier worker said: "It's hard to ask questions about something you don't fully understand". This is a paradoxical remark considering this person scored towards the top of this measure, yet they do not believe they "fully understand" the topic. Another top-scoring worker gave more insight on this apparent paradox, offering a possible explanation for this gap between the DNA measure and their ability to generate questions. This worker remarked how they "understand the nature of [DNA] fairly well", but what they do know "about DNA is pretty narrow". This could imply that some workers have a working knowledge of a topic, but their in-depth knowledge is limited in breadth. Another possible explanation for this potential paradox is that our measure for DNA expertise might be noisy. This measure is based on three closed-ended items and one open-ended item on the survey that explicitly mentions DNA, but we could consider including the handful of survey items that touched on the topic of genetics. Additional tuning of the DNA expertise measure is perhaps warranted.

5.4.3 Challenges: Workers thought the task was hard

Overall, crowd workers, regardless of expertise level, found the task of creating multiple questions of varying complexities quite challenging. Workers lamented about the task being "extremely difficult", "brutally tough", and "REALLY hard". A lack of knowledge in a particular topic was an issue mentioned by many workers. Also, "separating hard from easy" questions was another common issue. One worker said it was "hard to think about hard questions", while another remarked: "I couldn't think of any medium or hard questions". Yet others expressed uncertainty about what "qualifies as easy or hard" and difficulty in coming up with "something in between easy and hard." What was most surprising about these sentiments is that they were fairly uniform across all topics and expertise levels. Apparently, workers' perceived difficulty of the task was not a good predictor of their ability to generate suitable questions.

Remarkably, some workers "enjoyed" the challenge, commenting on the "fun" and "stimulating" nature of the task in light of its difficulty. One worker was "happy to be made to think", and another was thought it was "a lot of fun" even though the survey was "a lot harder than expected."

Interestingly, within the universe topic, a theme emerged. While many in the top tier did not find this task difficult, of those who did, what they found difficult was not their own lack of knowledge but the community's lack of knowledge. Five workers commented on how little is known about the universe, using phrases such as "we don't know", "a lot that is unknown", "so much...is unknown about it". One of these five workers actually found it "rather easy" to come up with questions because of this. In contrast to the top-tier theme, workers in the bottom tier generally found the task

difficult, primarily due to their own lack of knowledge (n=12) on the subject. They used phrases like "don't know much", "not knowledgeable", and "not very well versed". The difficulty for them rested on their personal lack of knowledge, whereas for the top tier, the difficulty was not personal but community-focused.

Some workers were surprised by how little they knew. One worker said they did not know "as many details...as I thought". Another echoed this sentiment: "I question how much I really know!" The survey helped these workers "realize there is a lot [they] don't know or remember," which "inspired" at least one of them "to do some more reading" and others to "study more", "learn more", and "start paying more attention".

Finally, the task of creating questions was actually easy for some workers. One worker summarized it by saying how it was "easy to conceptualize from previous knowledge." The sources of this "previous knowledge" were called out by various workers as "space documentaries", "previous science classes", TED talks on YouTube, and learning a lot about DNA growing up with a "mother [who] has a PhD in molecular genetics." Some workers acquired knowledge by being contemplative about certain topics, like having "a lot of thoughts about space" and being more interested in stem cells because of the "imminent arrival of a new baby soon." This contemplative state may have been spurned on by workers having "interest" in the topic or finding certain topics "very fascinating". Similar to what we found with perceived difficulty, workers' ability to generate suitable questions was not affected by the perceived ease of the task either.

5.5 Discussion

The goals of the Question Finding prototype was to identify specialized subcrowds, each having a range of expertise levels, differentiated by topical expertise,

and to solicit questions from the crowd that can be differentiated by complexity. We also wanted to better understand what difficulties people had in doing this. In this section, we discuss our findings in light of these goals. Integrated into this discussion are design implications for building a scalable, automated QF system.

5.5.1 Using topic-specific expertise measures

Findings from the study indicated the prototyped approach could identify numerous specialized subcrowds within the paid crowd, differentiated by topic-specific expertise. We have demonstrated this by creating and using topic-specific expertise measures based on a validated survey instrument. The expertise measures we found were for the biology and space domains, as well as the DNA, stem cell, and universe topics.

In a more general sense, the topical expertise measures (not space or biology) are quite close to what is considered as "discount" expertise metrics. Hung and Ackerman (2015) define discount expertise metrics as being easy to obtain and use. The topical expertise measures we used consisted of a handful of survey items each, but the scoring was not fully automated (disqualifying it from being a "discount" metric, in our opinion). Even though each topic included an open-ended response that we manually graded using a scoring rubric, this is not a fully automated expertise assessment. Teasing out additional closed-ended questions from the scoring rubric is one way to remove the need for manual scoring, but this fact-based approach runs the risk of losing some richness in the measurements that were introduced by the opinion-based criteria (discussed below).

Scoring the questions using a rubric is a problem nicely suited for crowdsourcing (Lasecki et al., 2014), where the crowd could use the rubric to provide subjective judgments about other workers' responses on open-ended items. This would allow for an

automated expertise measurement workflow while maintaining humans in the loop for added richness to the measurement. This would be a logical next step towards creating a scalable, automated QF system.

5.5.2 Differentiating question complexity

Being able to identify specialized crowds is interesting, but what does it enable? The ultimate goal of Question Finding is to eventually map the questions, differentiated by complexity, to suitable web content. The implied first step, which the prototype successfully addresses, is to generate questions of varying complexity. The findings demonstrate the crowd was able to do this.

The next step, then, is to scale this solution, which presents a challenge: determining the variations in question complexity was manually intensive. We tried two approaches: asking the workers to indicate the difficulty of their questions, and manually scoring the questions. We found that workers' judgments about the difficulty of the task were not a good predictor of their performance on the task. Thus, we had to rely on manual scoring of the questions by the research team. This is not scalable. With just 120 workers, there were almost 3,000 questions generated. Again, an interesting future direction for building a QF system is to use crowdsourcing to judge the quality of these questions.

The prototyped approach showed that predicting question complexity is useful, as we discuss next. But this capability is subject to a caveat about appropriately-scoped topics, which we discuss after that.

5.5.3 Strategic solicitation

The crowd experienced difficulties in differentiating hard from easy questions. This was more apparent for some topics (stem cells and solar system) than others (DNA and universe). By being able to identify worker expertise by topic, it is now possible to focus the efforts of the specialized crowds on specific difficulty levels. For example, with the DNA topic, top-tier workers were able to generate distinctly hard questions, but they produced easy and medium questions that were similar to the middle and bottom tier workers. This shows that the way to design QF systems is to focus the higher expertise workers on creating a variety of hard questions. As well, we know that workers with low expertise on the universe topic are much more reliable than higher-expertise workers at creating easy questions. Thus QF systems should solicit lower-tier workers to generate easy questions.

This strategic solicitation of questions affords QF system designers the flexibility to tradeoff task efficiency and question variety. Efficiencies are gained if designers chose to solicit, for example, only two hard questions from every expert-level worker. This would not only be cheaper and faster initially (as compared to our setup for this study), but it could have downstream effects if they rely on a separate mechanism to validate all the questions for suitability and complexity. In short, fewer questions to process would likely yield a cheaper and faster solution. If, on the other hand, the designer's goal is to maximize the variety of easy questions, then they might choose to solicit many more than two easy questions from every low-expertise worker. This strategically focused solicitation of work adds the flexibility for QF system designers to make tradeoffs.

5.5.4 Scoping the topic

As mentioned, an important finding in the evaluation of our QF approach was the need to consider the scope of the topic from which expertise is measured. We know that overly specific or complex topics (e.g., stem cells) lack widespread familiarity of the information, and this is a hindrance to the task of Question Finding. However, simply choosing topics that have widespread familiarity (e.g., solar system) is no guarantee of success, either. This is a classic problem in expertise finding on where to divide the topics and how narrow they should be (Merritt et al. 2016).

This points to the need for crowdsourcing system designers to assess a topic to determine if it is amenable to a diversity of expertise levels. Topic assessments would need to consider the scope of the topic (e.g., "molecular biology" versus "biology"), the widespread accessibility of information, and if there is a sufficient depth of information on the topic. In short, topic assessment needs to be a precursor to using mixed expertise in crowdsourcing.

5.5.5 Enriching the expertise measures

The QF prototype revealed some nuances about the combination of fact and opinion-based expertise measures, which was something we did not see in our studies of Escalier and Kurator. What we learned from this study is that open-ended questions can increase the resolution of the expertise measures. The fact-based (closed-ended) items on the survey were binary outcomes, making the scoring of those items a straightforward measure of percent correct. This tended to cause clumps of same scores for topical expertise measures that did not use many survey items. Using more closed-ended questions is one way to address this. However, we found that having even just one open-

ended question in the topic, scored on a non-binary scale, introduced more granularity into the measure's scale. Although these questions were scored using a rubric, we believe that human judgments about the correctness of the answers make this scoring an opinion-based one. Thus the inclusion of open-ended questions brings opinion-based criteria into the mix, which has the potential to increase the expertise measure's accuracy, particularly on complex items where there is a variance of correctness in an answer.

As well, opinion-based criteria could be used on top of fact-based ones. For example, if a multiple choice question (fact-based) was deemed especially difficult for most people, the scoring of the question could be weighted more heavily. The choice of this weighting is ultimately a subjective judgment about the relative difficulty of the question.

This problem of Question Finding lends itself to having subjective influence on how correctness is measured. This study revealed opportunities to inject opinion-based criteria into the expertise measures, and we found that subjective input added richness to the measurements.

5.6 Issues and Limitations

Our knowledge of the expertise measures we used is limited to how the 120 workers performed on the survey task. Will the crowd always produce similar scores for each measure? Perhaps we collected data on an unusual day where many high-expertise biology and space enthusiasts were using Mechanical Turk. To better understand the limitations, as well as the consistency, of these expertise measures, additional data collection instruments should be deployed over different days and times. This would

allow for a more generalizable index of expertise levels for the various topics we studied in this chapter.

What if, over time, information on stem cells or other measured topics becomes more widespread and understood? Presumably, crowd workers would score better on our expertise measures, possibly to the point of most workers appearing to have "uncommon" expertise. These measures would have to adjust over time to re-normalize to the crowd. This would keep our assessments anchored in the current reality, which makes statements such as "this person is a biology novice" or "that worker is a space expert" more trustworthy and enduring.

5.7 Conclusion

In this chapter, we introduced a prototype for an approach for Question Finding that was intended to meet two goals. The first goal was to identify specialized subcrowds, each having a range of expertise levels, differentiated by topical expertise. The QF prototype accomplished this goal by validating measures of expertise for multiple topics. The second goal was to solicit questions from the crowd that can be differentiated by complexity, and to determine if these differences were correlated to workers' expertise. The QF prototype accomplished this goal as well by demonstrating a way to leverage the information produced in the first goal. We found that identifying the topics and levels of expertise, according to our validated measures, is useful in predicting the difficulty level of questions generated by the crowd.

In a more general sense, we showed there are tangible benefits of using mixed expertise in this problem, which answers the third research question in this thesis. As well, the Question Finding problem overlaps the two problem areas this thesis has been

scoped to address. To identify specialized crowds, we established and tested several expertise measures for various topics and argued for their use as discount expertise metrics. These expertise measures used a combination of fact and opinion-based criteria, which helped us to better understand the nuances of the spectrum that the previous chapters did not uncover.

In the next chapter, we conclude this thesis through a discussion of the major contributions presented in this thesis, what it enables for researchers, the lessons learned, and the caveats.

Chapter 6. Conclusions

In this thesis, we proposed solutions to a limitation of current crowdsourcing approaches: not accounting for a range of expertise levels in the crowd. The current body of work in crowdsourcing does not systematically examine this, suggesting that researchers may not believe the benefits of using mixed expertise warrants the complexities of supporting it. This thesis presented two systems, Escalier and Kurator, as well as a study on Question Finding, to show that leveraging mixed expertise is a worthwhile endeavor because it materially benefits system performance, at scale, for various types of problems. We also demonstrated an effective technique, called expertise layering, to incorporate mixed expertise into crowdsourcing systems. Finally, we showed that leveraging mixed expertise enables researchers to use crowdsourcing to address new types of problems.

In the Introduction of this thesis, there were three research questions this thesis set out to answer:

- **RQ1**. Under what conditions, and to what extent, does using mixed expertise within a crowd materially benefit a crowdsourcing system at scale?
- **RQ2**. Under what conditions, and to what extent, is there benefit when using a mix of crowds, differentiated by types and levels of expertise, to solve a problem when the crowds work on similar tasks?

- **RQ3**. Under what conditions, and to what extent, is there benefit when using a mix of subcrowds within a crowd, differentiated by expertise?

This thesis has answered these questions. In our study of the Escalier system, we investigated expertise at scale for fact-based expertise criteria, and there were some feedback loops and mixed expertise leveraged from the same crowd. In a deeper investigation of feedback loops and mixed expertise in different crowds, particularly in a domain where opinion-based expertise criteria was at play, we built and studied the Kurator system. Kurator represents a workable solution that uses opinion-based expertise criteria. Using Kurator, we started to identify specialized crowds, but we wanted to investigate more fully how we might identify and leverage specialized crowds, particularly subcrowds within a crowd, differentiated by expertise. Also, we wanted to better understand how a system might work in the middle of the fact-opinion spectrum of expertise criteria. This led to our study on Question Finding, which demonstrated that we were able to identify specialized subcrowds who are differentiated by expertise. We created several expertise measures for various topics. These measures used a combination of fact and opinion-based criteria, which helped us to better understand the nuances of the spectrum.

6.1 General Conclusions

There are many conclusions to be drawn from the work presented in this thesis. Here we walk through many of the salient conclusions organized into a "path" for crowdsourcing system designers to follow. Our articulation of the path itself is arguably a contribution to the crowdsourcing literature, but our intent is simply to walk the reader along a cogent narrative of the areas we believe we understand better because of this

thesis. Borrowing from military command and control doctrine, we use the OODA Loop--Observe, Orient, Decide, Act (Boyd, 1987)--as a mental model for this path, except we include a fifth step: Leverage.

6.1.1 Observe: A mix of expertise exists in the crowd.

As the starting point on the path for building crowdsourcing systems to leverage mixed expertise, crowdsourcing system designers must first "observe" that a mix of expertise even exists in the crowd. The literature review in Chapter 2 made it clear that human expertise likely exists on a continuum. The work in this thesis has given us more reason to believe this is true. Kurator taught us that there exists in the crowd some latent expertise for predicting how parents make decisions about sentimental audio recordings. If the "crowd" is extended to include family members, a mix of expertise is plain to see in the combination of "expert" parents, less expert family and friends, crowd workers with some expertise, and crowd workers with little expertise. Question Finding further developed this belief by showing us there are specialized subcrowds with various levels of scientific expertise, specifically within the topics of DNA, the universe, the solar system, and space and biology, in general.

6.1.2 Orient: Choose the "right" topic and type of expertise.

Knowing there is a mix of expertise is only helpful if designers know how to "orient" the expertise topics and types that their system will use. Our notion of "right" means there are some topics and types that are better choices than others, but there is likely not a single correct topic or type of expertise. The study of Question Finding illuminated the importance of scoping a topic appropriately, paying careful attention to the breadth of the topic, the widespread accessibility of information, and if there is a

sufficient depth of information on the topic. As well, throughout this thesis, we have seen varying types of expertise, from fact-based (Escalier) to opinion-based (Kurator). A combination of the two (Question Finding) proved useful, too. Designers ought to consider how well-structured or ill-structured their domain is (Voss and Wiley, 2006), which will guide their choice of expertise type (see Chapter 2 for more details).

6.1.3 Decide: There are ways to identify some of that expertise.

Equipped with a chosen expertise topic and type, designers must decide how they will identify expertise in that topic. This thesis partially worked out several ways to identify a mix of expertise in the crowd. Kurator tracked the crowd's expertise as a performance-based measure, possibly to be used later in weighted voting. The Question Finding study borrowed a validated survey instrument, added to it, and used the subcomponents of the survey for topic-based expertise measures. That study also suggested it might be useful to crowd-source subjective judgments about question complexity. Designers can use other established metrics, similar to how Escalier used social network analysis (e.g., Zhang et al. 2007) and web browsing-based discount expertise metrics (Hung and Ackerman, 2015).

6.1.4 Act: Build an expertise-aware system without having expertise at the start.

Designers may find themselves desiring to build a system to incorporate a topic and/or type of expertise that has not already been established. This thesis demonstrates at least one way to "act" on that desire to build without the need to use expertise right away. Expertise layering is a useful guiding concept for system builders, and we demonstrate its effectiveness with Escalier and Kurator. Expertise layering avoids "expertise cold start", which can be thought of as the absence of available expertise at system initialization.

Escalier is an example of a social navigation system that can leverage mixed expertise when the expertise becomes available, and Kurator is similarly built to be effective "out of the box" but can improve as experts contribute to the tasks.

6.1.5 Leverage. Expertise is additive

Finally, even with a workable crowdsourcing solution that successfully accounts for mixed expertise, designers should remember there is more benefit to be gained if they continually "leverage" new data, new expertise, and new expertise measurement algorithms or approaches. This is possible because expertise is additive. The literature on expertise finder systems taught us that, as systems evolve over time, more and more data and algorithms were added into systems to continually refine the expertise measures. Thus, where data is available and relevant, designers should continue to leverage it.

This path of system design considerations for incorporating mixed expertise is surely not the only path. However, the work in this thesis has demonstrated this path is workable, and we offer it as one way we believe is helpful to crowdsourcing system designers.

6.2 Limitations

There are many limitations to the studies presented in this thesis. Here we draw the reader's attention to two salient ones.

First, the systems and studies in this thesis primarily used Mechanical Turk, which means our results are generalizable only to paid microtasking environments. Paid *micro*tasking environments are designed for *micro* tasks. If a task is really trivial, then maybe mixed expertise does not matter. Perhaps mixed expertise only matters for tasks beyond a certain level of complexity. Crowdsourcing systems leveraging mixed

expertise in paid microtasking environments will likely have to face a tension between breaking a problem into trivial tasks while also maintaining enough complexity to maximize the value of using human input. We did not examine this tension, so we do not know the effects it will have on mixed expertise crowdsourcing system design.

Second, in our study of Kurator, we speculate about the benefits of using weighted crowd voting. Tracking crowd workers' expertise is only helpful if the expertise can be leveraged, and weighting inputs from known[13] workers is a straightforward way to account for their expertise. Although we assume this is a workable approach, we do not examine this idea of weighted voting. Doing so, and showing it is indeed workable, would bolster the strength of our argument that leveraging mixed expertise is worthwhile.

6.3 Future Direction

This thesis brings to light several areas where future research may be fruitful. In addition to the areas we discussed in the previous section, here we discuss two more that deserve attention.

First, it is important to acknowledge that not all problems are better solved by mixed expertise. When a crowd of experts is accessible and practical to use, as is the case for Master[14] image labelers on Mechanical Turk, then it makes little sense to attempt to incorporate less-expert workers. This is only one example, but there are likely many more. This thesis has not surveyed the current state of the art to uncover problem areas

---

[13] We do not mean their identity is known; we mean to say their expertise is known.

[14] Visit this website for more an explanation of what a Master turker is: https://www.mturk.com/mturk/help?helpPage=worker#what_is_master_worker

and tasks where mixed expertise may be moot. It would be useful to have this knowledge, and examining the progression of problems where experts have become more accessible over time is a warranted future direction for this research.

Second, our review of the CSCW prior literature on expertise finder system taught us that social interaction data is helpful in expertise measurements, but we did not attempt to address this data gap. Although microtasking environments, like Mechanical Turk, maintain a certain level of anonymity, researchers have begun to build self-coordinating crowdsourcing infrastructure (e.g., Apparition, by Lasecki et al. 2015). Crowdsourcing systems that rely on interaction and collaboration between workers will likely have useful metadata for expertise measurements. To gain a better understanding of the nature of expertise in crowds, it is important to study crowdsourcing systems that generate interactional data and, perhaps, to incorporate more social data into expertise measures.

Appendices

Appendix A

User Search and Voting Algorithms in Escalier

Accuracy Factor:

In the model of user behavior, the highest level of expertise makes edits to modules based on the same distribution used to determine the Stable set from the Canonical set. As the levels of expertise decrease, the distribution curve flattens out, slowly approaching a uniform distribution of numbers of edits. The lowest level of expertise actually uses a uniform distribution, where each selection of the number of edits to modules is equiprobable. In Figure 12, one simulation run's sampling of the five distributions validates that the distribution curves are flattening out as the expertise decreases.

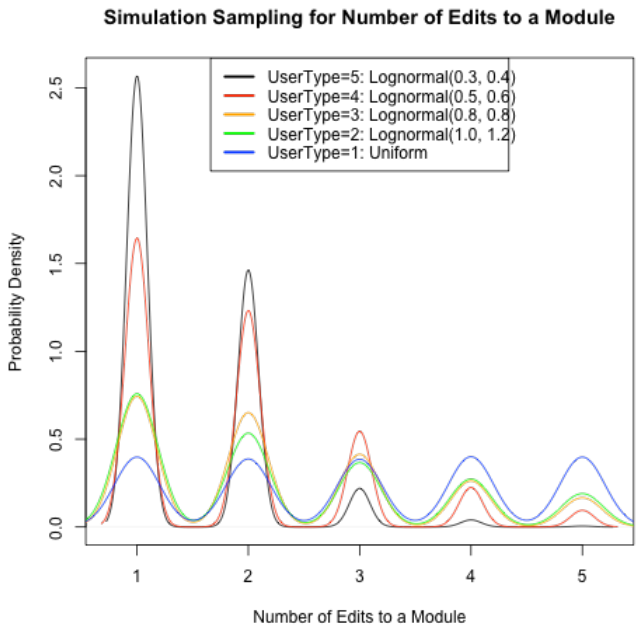


Figure 12. Simulation Sampling for Number of Edits to a Module

User Search Algorithm:

1) User is probabilistically assigned a UserType based on a random sample drawn uniformly, as shown in Table 9.
    a. The five UserTypes correspond to five user expertise levels. UserType 1 has the least expertise, and UserType 5 has the most expertise.

Table 9. Probabilistic Selection of 5 User Types (randomly selected values [0,1])

UserType	1	2	3	4	5
Random Sample	≤ 0.4748	≤ 0.7354	≤ 0.8784	≤ 0.9569	> 0.9569

2) Randomly select the starting configuration from the IS set.
3) Choose the number of search attempts based on the Persistence Factor.
    a. Number of attempts = 3 x UserType, where UserType is an integer between 1 and 5, inclusive.
    b. For example, relative novices (UserType=1) will make 3 attempts, and relative experts (UserType=5) will make 15 attempts.
    c. I also implemented Persistence Factors of 1 and 2 to use in a sensitivity analysis.
4) For each search attempt:
    a. Modify the starting configuration using the lognormal distribution of number of modules to edit, the lognormal distribution of number of edits to each module, and the lognormal distribution of new value selections.
        i. The distribution of number of edits to each module is determined by the user's Accuracy Factor.

b. User performs a Vote() operation on the Edited Configuration

c. If the result of the Vote() operation is "stable", then end the search

i. Else if all search attempts have been made, then end the search

ii. Else, go to step 3.

User Voting Algorithm:

1) Probabilistically vote ("stable" or "not stable") on the configuration at hand.

a. The vote equals the *actual stability* of the configuration according to the probabilities based on the UserType, as shown in Table 10. The *actual stability* of the configuration is determined by the configuration's membership in the Stable set, where $C_i \in S$ means it is "stable"; otherwise, it is "not stable".

b. The probability starts at 0.6 because I decided domain novices should perform better than sheer luck (>0.5). I also decided domain experts very rarely make incorrect assessments but, being human, are not perfect (<1.0).

Table 10. Vote Probability Based on UserType

UserType	1	2	3	4	5
Prob(vote = actual stability)	0.6	0.7	0.8	0.9	0.99

2) Increment or decrement the configuration's stability rating depending on the result of the vote ("stable" or "not stable", respectively).

Table 11. Weighted Rating Based on UserType

UserType	1	2	3	4	5
Rating	0.2	0.3	0.4	0.6	0.8

a. If Escalier L3 layer is enabled, then the increment/decrement value is weighted based on the user's expertise level, as shown in Table 10.

b. Otherwise, it is equally weighted at 0.25.

Appendix B

Canonical and Stable Generation Algorithms in Escalier

Canonical-generation Algorithm:

For each configuration in the Canonical set, the integer values for each of the elements of its vector are selected from a lognormal distribution of value selections for each property. We chose a lognormal distribution of values based on a manual analysis of the Linux data, where I observed that the same properties between configurations mostly used the same one or two values. A less common but still noteworthy observation we made was the opposite of this lognormal distribution: there did not appear to be a favoring towards any one value but, rather, a fairly equal distribution of values (possibly a uniform distribution). These two observations are intuitive, as the former indicates properties where only one or two values are important, and the latter indicates properties where every value is important. Thus we initially use the lognormal distribution of value selections in order to cluster the Canonical set in the configuration space based on preferential attachment. However, we later modify the simulation using a normal distribution of value selections to explore the effect of the distribution of values per property.

Stable-generation Algorithm:

1) Select a random Canonical configuration.
2) Determine the *number of modules* to edit by sampling from the lognormal distribution.
3) For each module to edit, determine the *number of edits* to make to the module by sampling from the lognormal distribution.
4) For each edit to make within a module, *select a new value* by sampling from the lognormal distribution.
    - The new value cannot be equal to the current value.
5) Append the modified configuration to the Stable set
    - If Stable set is fully populated, then start simulation; else go to Step 1

The following bullets are further discussions about the Stable-generation algorithm:

- In Step 2:
    - Even though I do not directly calculate how many *modules* are edited from one configuration to the next, my empirical observations reveal that at least some modules remain the same.
    - This lognormal distribution probabilistically limits the editing to only a few modules most of the time, which has support in the data and by personal experience. However, I performed a sensitivity analysis using a uniform distribution to compare the effects of non-clustering of the topology.
- In Step 3:

- Section 5.3 discusses *two* distributions for intra-module Hamming distances: lognormal and normal. Thus I performed a sensitivity analysis using a normal distribution as well.

- In Step 4:
  - I also performed a sensitivity analysis using a uniform distribution.

Appendix C

Question Finding: Example Questions from the Crowd

Easy Questions

- Does everyone have DNA?
- Do animals have DNA?
- What does DNA stand for?
- Where is DNA found?
- How big is the universe?
- How many galaxies are there in the universe?
- Are there other planets in the universe?
- What is the largest galaxy in the universe?

Medium Questions

- How does DNA function?
- What causes DNA to degrade over time?
- What happens when there are mutations in DNA?
- How is DNA imparted to offspring?
- What is the temperature of the universe?
- At what rate is the universe expanding?
- Is it possible that at some point the universe runs out of outward momentum and starts to collapse in on itself again?
- How can the big bang form matter and energy out of nothing?

Hard Questions

- How does DNA replicate?
- Does DNA control gene expression?
- Why is the DNA molecule shaped like a helix?
- What translates DNA into messenger RNA?
- Where is the missing dark matter in our universe?
- How are space and time connected?
- How do blue and red shift contribute to our knowledge of the universe?
- What percent of the matter in the universe is contained in supermassive black holes?

References

Ackerman, M.S., 2000. The intellectual challenge of CSCW: the gap between social requirements and technical feasibility. *Human-computer interaction*,*15*(2), pp.179-203.

Ackerman, Mark S., James S. Boster, Wayne G. Lutters, and David W. McDonald (2003). "Who's There? The Knowledge-Mapping Approximation Project." In M. Ackerman, V. Pipek, and V. Wulf (Eds.), *Beyond Knowledge Management: Sharing Expertise.* Cambridge, MA: MIT Press, 159-178.

Ackerman, M.S., Dachtera, J., Pipek, V. and Wulf, V., 2013. Sharing knowledge and expertise: The CSCW view of knowledge management. *Computer Supported Cooperative Work (CSCW)*, *22*(4-6), pp.531-573.

Adomavicius, G. and Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Engineering*, 17, 6, 734-749.

Arnold, H.J. and Feldman, D.C., 1981. Social desirability response bias in self-report choice situations. *Academy of Management Journal*, *24*(2), pp.377-385.

Bandini, S., Manzoni, S., & Vizzari, G. 2004. Situated cellular agents: A model to simulate crowding dynamics. *IEICE Trans. on Information and Systems*, *87*(3), 669-676.
Barto, A. 1998. Reinforcement learning: An introduction. MIT press.

Barto, A. 1998. *Reinforcement learning: An introduction*. MIT press.

Becks, Andreas, Tim Reichling, and Volker Wulf. (2004). "Expert Finding: Approaches to Foster Social Capital." In M. Huysman and V. Wulf (Eds.), *Social Capital and Information Technology.* Cambridge, MA, 333-354.

Berger, P.L. and Luckmann, T., 1991. *The social construction of reality: A treatise in the sociology of knowledge* (No. 10). Penguin UK.

Bernard, H.R., 2011. *Research methods in anthropology: Qualitative and quantitative approaches*. Rowman Altamira.

Bernstein, M. S., Ackerman, M. S., Chi, E. H., & Miller, R. C. 2011a. The trouble with social computing systems research. *Proc. CHI EA '11.* pp. 389-398.

Bernstein, M.S., Brandt, J., Miller, R.C. and Karger, D.R., 2011b. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In*Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 33-42). ACM.

Bernstein, M.S., Little, G., Miller, R.C., Hartmann, B., Ackerman, M.S., Karger, D.R., Crowell, D. and Panovich, K., 2010a, October. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (pp. 313-322). ACM.

Bernstein, M.S., Tan, D., Smith, G., Czerwinski, M. and Horvitz, E., 2010b. Personalization via friendsourcing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *17*(2), p.6.

Bigham, J.P., Bernstein, M.S. and Adar, E., 2015. Human-Computer Interaction and Collective Intelligence. *Handbook of Collective Intelligence*, p.57.

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., 2010, October. VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (pp. 333-342). ACM.

Boyd, J. (1987). A discourse on winning and losing. Maxwell Air Force Base, AL: Air University Library Document No. M-U 43947 (Briefing slides: http://www.au.af.mil/au/awc/awcgate/boyd/osinga_boydconf07_copyright2007.pdf)

Bozzon, Alessandro, Marco Brambilla, Stefano Ceri, Matteo Silvestri, and Giuliano Vesci (2013). Choosing the right crowd: expert finding in social networks. In *Proceedings of the 16th International Conference on Extending Database Technology* (EDBT '13). ACM, New York, NY, USA, 637-648.

Ceroni, A., Solachidis, V., Fu, M., Kanhabua, N., Papadopoulou, O., Niederée, C. and Mezaris, V., 2015a. Investigating human behaviors in selecting personal photos to preserve memories. In *Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on* (pp. 1-6). IEEE.

Ceroni, A., Solachidis, V., Niederée, C., Papadopoulou, O., Kanhabua, N. and Mezaris, V., 2015b. To Keep or not to Keep: An Expectation-oriented Photo Selection Method for Personal Photo Collections. In*Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (pp. 187-194). ACM.

Chan, J., Dang, S. and Dow, S.P., 2016. Improving Crowd Innovation with Expert Facilitation. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*.

Chang, S., Kumar, V., Gilbert, E. and Terveen, L.G., 2014, February. Specialization, homophily, and gender in a social curation site: findings from pinterest. In *Proceedings*

of the 17th ACM conference on Computer supported cooperative work & social computing (pp. 674-686). ACM.

Chase, W.G. and Simon, H.A., 1973. The mind's eye in chess.

Chau, D., Nachenberg, C., Wilhelm, J., Wright, A. and Faloutsos, C. 2011. Polonium: Tera-scale graph mining and inference for malware detection. *Proc. SDM '11*

Chi, M.T., 2006. Two approaches to the study of experts' characteristics. In K. A. Ericsson, N. Charness, P. Feltovich, and R. R. Hoffman, R. R. (Eds.). *Cambridge handbook of expertise and expert performance* (pp. 21-30). Cambridge, UK: Cambridge University Press.

Chilton, L.B., Kim, J., André, P., Cordeiro, F., Landay, J.A., Weld, D.S., Dow, S.P., Miller, R.C. and Zhang, H., 2014, April. Frenzy: collaborative data organization for creating conference sessions. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (pp. 1255-1264). ACM.

Cohen, Jacob. "A coefficient of agreement for nominal scales." *Educational and psychological measurement* 20, no. 1 (1960): 37-46.

Collins, H., 2006. Three Dimensions of Expertise. In J.W. Treem and P.M. Leonardi (Eds.). *Expertise, Communication, and Organizing* (pp. 60-78). Oxford, UK: Oxford University Press.

Cusano, C. and Santini, S., 2014. With a little help from my friends. *Multimedia tools and applications*, *70*(2), pp.1033-1048.

Dieberger A, Dourish P, Höök K, Resnick P, Wexelblat A. Social navigation: techniques for building more usable systems. interactions. 2000 Nov 1;7(6):36-45.

DiGioia P, Dourish P. Social navigation as a model for usable security. InProceedings of the 2005 symposium on Usable privacy and security 2005 Jul 6 (pp. 101-108). ACM.

Donaldson, S.I. and Grant-Vallone, E.J., 2002. Understanding self-report bias in organizational behavior research. *Journal of Business and Psychology*,*17*(2), pp.245-260.

Dow, S., Kulkarni, A., Klemmer, S. and Hartmann, B., 2012, February. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1013-1022). ACM.

Dreyfus, Stuart E. and Dreyfus, Hubert L. 1980. *A five-stage model of the mental activities involved in directed skill acquisition (No. ORC-80-2)*. California University Berkeley Operations Research Center, 1980.

Eaton, John W., David Bateman, and Søren Hauberg (2009). GNU Octave version 3.0.1 manual: a high-level interactive language for numerical computations. CreateSpace Independent Publishing Platform. ISBN 1441413006, http://www.gnu.org/software/octave/doc/interpreter/

Ehrlich, Kate (2003). "Locating Expertise: Design Issues for an Expertise Locator System." In M. Ackerman, V. Pipek, and V. Wulf (Eds.), *Beyond Knowledge Management: Sharing Expertise.* Cambridge, MA: MIT Press, 137-158.

Ericsson, K. A. 2006a. An introduction to Cambridge Handbook of Expertise and Expert Performance: its development, organization, and content. In K. A. Ericsson, N. Charness, P. Feltovich, and R. R. Hoffman, R. R. (Eds.). *Cambridge handbook of expertise and expert performance* (pp. 3-19). Cambridge, UK: Cambridge University Press.

Ericsson, K.A., Charness, N., Feltovich, P.J. and Hoffman, R.R. eds., 2006b. *The Cambridge handbook of expertise and expert performance*. Cambridge University Press.

Eslami, M., Rickman, A., Vaccaro, K., Aleyasen, A., Vuong, A., Karahalios, K., Hamilton, K. and Sandvig, C., 2015, April. I always assumed that I wasn't really that close to [her]: Reasoning about Invisible Algorithms in News Feeds. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 153-162). ACM.

Farrell, Stephen, Tessa Lau, Stefan Nusser, Eric Wilcox, and Michael Muller (2007). "Socially augmenting employee profiles with people-tagging." In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pp. 91-100. ACM.

Foner. Leonard N. (1997). Yenta: a multi-agent, referral-based matchmaking system. In *Proceedings of the first international conference on Autonomous agents* (AGENTS '97). ACM, New York, NY, USA, 301-307.

Foner, Leonard N., and I. Barry Crabtree (1997). "Multi-agent matchmaking." In *Software Agents and Soft Computing Towards Enhancing Machine Intelligence*, pp. 100-115. Springer Berlin Heidelberg.

Fort, Karën, Gilles Adda, and K. Bretonnel Cohen. 2011. Amazon Mechanical Turk: gold mine or coal mine? *Computational Linguistics* **37**(2): 413–420.

Freyne, Jill, Rosta Farzan, Peter Brusilovsky, Barry Smyth, and Maurice Coyle. 2007. "Collecting community wisdom: integrating social search & social navigation." In *Proceedings of the 12th international conference on Intelligent user interfaces*, pp. 52-61. ACM.

Gilbert, N., & Troitzsch, K. 2005. *Simulation for the social scientist*. McGraw-Hill International.

Goecks J, Edwards WK, Mynatt ED. Challenges in supporting end-user privacy and security management with social navigation. InProceedings of the 5th Symposium on Usable Privacy and Security 2009 Jul 15 (p. 5). ACM.

Guldogan, E., Kangas, J. and Gabbouj, M., 2013, January. Personalized representative image selection for shared photo albums. In *Computer Applications Technology (ICCAT), 2013 International Conference on* (pp. 1-4). IEEE.

Gulotta, R., Sciuto, A., Kelliher, A. and Forlizzi, J., 2015, April. Curatorial Agents: How Systems Shape Our Understanding of Personal and Familial Digital Information. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 3453-3462). ACM.

Hämäläinen, A., Meinedo, H., Tjalve, M., Pellegrini, T., Trancoso, I. and Dias, M.S., 2014. Improving Speech Recognition through Automatic Selection of Age Group–Specific Acoustic Models. In *Computational Processing of the Portuguese Language* (pp. 12-23). Springer International Publishing.

Hill, W.C., Hollan, J.D., Wroblewski, D. and McCandless, T., 1992, June. Edit wear and read wear. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 3-9). ACM.

Hinds, Pamela J., and Jeffrey Pfeffer (2003). "Why organizations don't 'know what they know': Cognitive and motivational factors affecting the transfer of expertise." In M. Ackerman, V. Pipek, and V. Wulf (Eds.), *Beyond Knowledge Management: Sharing Expertise*. Cambridge, MA: MIT Press, 3-26.

Hoffman, R.R., 1998. How can expertise be defined? Implications of research from cognitive psychology. *Exploring expertise*, pp.81-100.

Horowitz, Damon, and Sepandar D. Kamvar. 2010. "The anatomy of a large-scale social search engine." In *Proceedings of the 19th international conference on World wide web*, pp. 431-440. ACM.

Howe, J. Crowdsourcing: A definition. (2006). http://crowdsourcing.typepad.com/cs/2006/06/crowdsou rcing_a.html

Hu, C., Bederson, B.B., Resnik, P. and Kronrod, Y., 2011, May. MonoTrans2: a new human computation system to support monolingual translation. In*Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1133-1136). ACM.

Huang, T.H.K., Lasecki, W.S. and Bigham, J.P., 2015, September. Guardian: A Crowd-Powered Spoken Dialog System for Web APIs. In *Third AAAI Conference on Human Computation and Crowdsourcing*.

Huh, Jina, Newman, Mark W., and Ackerman, Mark S. Supporting collaborative help for individualized use. In *CHI '11: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), ACM, 3141-3150.

Hung, P.Y. and Ackerman, M.S., 2015. Discount Expertise Metrics for Augmenting Community Interaction. In *Proceedings of the Work-In-Progress Track of the 7th International Conference on Communities and Technologies* (Vol. 12, No. 1, pp. 43-52).

Jones, Jasmine, Merritt, David T., Ackerman, Mark S. 2016. KidKeeper: Design for capturing candid memories for parents of young children. *In submission to CSCW '17*.

Jung, H.J. and Lease, M., 2015, September. Modeling Temporal Crowd Work Quality with Limited Supervision. In *Third AAAI Conference on Human Computation and Crowdsourcing*.

Kajino, H., Baba, Y. and Kashima, H., 2014, May. Instance-Privacy Preserving Crowdsourcing. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Kao, Wei-Chen, Duen-Ren Liu, and Shiu-Wen Wang (2010). Expert finding in question-answering websites: a novel hybrid approach. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (SAC '10). ACM, New York, NY, USA, 867-871.

Kim, J., Cheng, J. and Bernstein, M.S., 2014, February. Ensemble: exploring complementary strengths of leaders and crowds in creative collaboration. In*Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing* (pp. 745-755). ACM.

Kittur, A., & Kraut, R. E. 2008. Harnessing the wisdom of crowds in wikipedia: quality through coordination. *Proc. CSCW '08* (pp. 37-46). ACM.

Kittur, Aniket, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton, 2013. "The future of crowd work." In *Proceedings of the 2013 conference on Computer supported cooperative work*, pp. 1301-1318. ACM.

Kokkalis, N., Köhn, T., Pfeiffer, C., Chornyi, D., Bernstein, M.S. and Klemmer, S.R., 2013, February. EmailValet: managing email overload through private, accountable crowdsourcing. In *Proceedings of the 2013 conference on Computer supported cooperative work* (pp. 1291-1300). ACM.

Kulkarni, A., Narula, P., Rolnitzky, D., & Kontny, N. 2014. Wish: Amplifying Creative Ability with Expert Crowds. *Proc. HCOMP '14*.

Kumar, P. and Schoenebeck, S., 2015, February. The modern day baby book: Enacting good mothering and stewarding privacy on facebook. In*Proceedings of the 18th ACM*

Conference on Computer Supported Cooperative Work & Social Computing (pp. 1302-1312). ACM.

Larkin, J., McDermott, J., Simon, D.P. and Simon, H.A., 1980. Expert and novice performance in solving physics problems. *Science*, *208*(4450), pp.1335-1342.

Lasecki, W.S., Gordon, M., Koutra, D., Jung, M.F., Dow, S.P. and Bigham, J.P., 2014, October. Glance: Rapidly coding behavioral video with the crowd. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (pp. 551-562). ACM.

Lasecki, W.S., Kim, J., Rafter, N., Sen, O., Bigham, J.P. and Bernstein, M.S., 2015, April. 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* (pp. 1925-1934). ACM.

Lasecki, W., Miller, C., Sadilek, A., Abumoussa, A., Borrello, D., Kushalnagar, R. and Bigham, J., 2012, October. Real-time captioning by groups of non-experts. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (pp. 23-34). ACM.

Lasecki, W.S., Murray, K.I., White, S., Miller, R.C. and Bigham, J.P., 2011, October. Real-time crowd control of existing interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 23-32). ACM.
Lasecki, W.S., Teevan, J. and Kamar, E., 2014, February. Information extraction and manipulation threats in crowd-powered systems. In*Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing* (pp. 248-256). ACM.

Lasecki, W. S., Thiha, P., Zhong, Y., Brady, E., & Bigham, J. P. 2013. Answering visual questions with conversational crowd assistants. *Proc. ASSETS '13.* (p. 18). ACM.

Law, E., Dalton, C., Merrill, N., Young, A. and Gajos, K.Z., 2013, March. Curio: A Platform for Supporting Mixed-Expertise Crowdsourcing. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Le, D., and Mower Provost, E. 2013. Emotion recognition from spontaneous speech using hidden markov models with deep belief networks. In *Automatic Speech Recognition and Understanding*, 216–221. IEEE.

Lease, M., Hullman, J., Bigham, J.P., Bernstein, M.S., Kim, J., Lasecki, W., Bakhshi, S., Mitra, T. and Miller, R.C., 2013. Mechanical turk is not anonymous. *Available at SSRN 2228728*.

Lee, J. W., Helal, A., Sung, Y., & Cho, K. 2013. A context-driven approach to scalable human activity simulation. *Proc. SIGSIM PADS '13.* pp. 373-378. ACM.

Lehmann, A. C. and Gruber, H., 2006. Music. In K. A. Ericsson, N. Charness, P. Feltovich, and R. R. Hoffman, R. R. (Eds.). *Cambridge handbook of expertise and expert performance* (pp. 457-470). Cambridge, UK: Cambridge University Press.

Li, J., Lim, J.H. and Tian, Q., 2003, December. Automatic summarization for personal digital photos. In *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on* (Vol. 3, pp. 1536-1540). IEEE.

Lin, Ching-Yung, Kate Ehrlich, Vicky Griffiths-Fisher, and Christopher Desforges (2008). "Smallblue: People mining for expertise search." *MultiMedia, IEEE*15, no. 1, 78-84.

MacDougall, M. H. 1970. Computer system simulation: An introduction. ACM Computing Surveys (CSUR), 2(3), 191-209.

Marshall, Catherine C (2007). How people manage personal information over a lifetime. Personal information management, pp. 57-75.

Marshall, Catherine C, Sara Bly, and Francoise Brun-Cottan (2006). The long term fate of our digital belongings: Toward a service model for personal archives. Proceedings of the Archiving Conference, pp. 25-30.

Martin, David, Benjamin V. Hanrahan, Jacki O'Neill, and Neha Gupta. 2014. Being a Turker. *Proc. CSCW '14*: 224–235.

McDonald, David W., and Mark S. Ackerman (2000). "Expertise recommender: a flexible recommendation system and architecture." In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pp. 231-240. ACM.

McDonald, David W. (2001). Evaluating Expertise Recommendation. In *Proceedings of the 2001 International ACM Conference on Supporting Group Work*, ACM Press, New York, 214-223.

Meinedo, H. and Trancoso, I., 2011. Age and gender detection in the I-DASH project. *ACM Transactions on Speech and Language Processing (TSLP)*,*7*(4), p.13.

Merritt, D.T., Ackerman, M.S., & Hung, P.Y. 2016. Expertise Finding. In J.W. Treem and P.M. Leonardi (Eds.). *Expertise, Communication, and Organizing* (pp. 100-122). Oxford, UK: Oxford University Press.

Milland, Kristy ("spamgirl"). 2014. The myth of low cost, high quality on Amazon's Mechanical Turk. *Turker Nation*, 30 Jan 2014.

Miller, J.D., 1998. The measurement of civic scientific literacy. *Public understanding of science*, *7*(3), pp.203-223.

Miller, J.D., 2010. The conceptualization and measurement of civic scientific literacy for the twenty-first century. *Science and the educated American: A core component of liberal education*, *136*.

Miller, J.D., 2012a. 13 The Sources and Impact of Civic Scientific Literacy. *The culture of science: How the public relates to science across the globe*, *15*, p.217.

Miller, J., 2012b. What Colleges and Universities Need to Do to Advance Civic Scientific Literacy and Preserve American Democracy. *Liberal Education*, p.29.

Mooney, C. Z. *Monte carlo simulation*. Sage, 1997.

Munger, Tyler, and Jiabin Zhao (2014). "Automatically identifying experts in on-line support forums using social interactions and post content." In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 930-935. IEEE.

Nejdl, W. and Niederee, C., 2015. Photos to Remember, Photos to Forget.*MultiMedia, IEEE*, *22*(1), pp.6-11.

Nguyen, A.T., Wallace, B.C. and Lease, M., 2015, September. Combining Crowd and Expert Labels using Decision Theoretic Active Learning. In *Third AAAI Conference on Human Computation and Crowdsourcing*.

Noice, H. and Noice, T., 2006. Artistic performance: Acting, ballet and contemporary dance. In K. A. Ericsson, N. Charness, P. Feltovich, and R. R. Hoffman, R. R. (Eds.). *Cambridge handbook of expertise and expert performance* (pp. 489-503). Cambridge, UK: Cambridge University Press.

Obrador, Pere, Rodrigo De Oliveira, and Nuria Oliver. 2010. "Supporting personal photo storytelling for social albums." In *Proceedings of the international conference on Multimedia*, pp. 561-570. ACM, 2010.

Oleksik, G. and Brown, L.M. 2008. Sonic gems: exploring the potential of audio recording as a form of sentimental memory capture. *Proc. BCS-HCI 2008*, Vol. 1, British Computer Society, 163-172.

Organisciak, Peter, Jaime Teevan, Susan Dumais, Robert C. Miller, and Adam Tauman Kalai. 2014. "A Crowd of Your Own: Crowdsourcing for On-Demand Personalization." In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Petroski, H., 1994. *Design paradigms: Case histories of error and judgment in engineering*. Cambridge University Press.

Quinn, A.J. and Bederson, B.B., 2011, May. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1403-1412). ACM.

Rader, E. and Gray, R., 2015, April. Understanding user beliefs about algorithmic curation in the Facebook news feed. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 173-182). ACM.

Reichling, Tim, Kai Schubert, and Volker Wulf (2005). Matching Human Actors Based on Their Texts: Design and Evaluation of an Instance of the ExpertFinding Framework. In *Proceedings of the International Conference on Supporting Group Work (Group'05)*, pp. 61–70.

Retelny, D., Robaszkiewicz, S., To, A., Lasecki, W.S., Patel, J., Rahmati, N., Doshi, T., Valentine, M. and Bernstein, M.S., 2014, October. Expert crowdsourcing with flash teams. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (pp. 75-85). ACM.

Sadjadi, S.O. and Hansen, J.H., 2013. Unsupervised speech activity detection using voicing measures and perceptual spectral flux. *Signal Processing Letters, IEEE*, *20*(3), pp.197-200.

Schuller, B.; Batliner, A.; Steidl, S.; and Seppi, D. 2011. Recog- nising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge. *Speech Communication* 53(9):1062–1087.

Simko, J. and Bieliková, M., 2011, December. Games with a purpose: User generated valid metadata for personal archives. In *Semantic Media Adaptation and Personalization (SMAP), 2011 Sixth International Workshop on* (pp. 45-50). IEEE.

Snow, R., O'Connor, B., Jurafsky, D. and Ng, A.Y., 2008, October. Cheap and fast---but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 254-263). Association for Computational Linguistics.

Sorokin, A. and Forsyth, D., 2008. "Utility data annotation with Amazon Mechanical Turk", *CVPRW*, 2008, 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops 2008, pp. 1-8.

Star, Susan Leigh, and Anselm Strauss (1999). "Layers of silence, arenas of voice: The ecology of visible and invisible work." *Computer supported cooperative work (CSCW)* 8, no. 1-2: 9-30.

Streeter, Lynn A., and Karen E. Lochbaum, 1988. "Who knows: a system based on automatic representation of semantic structure." In *RIAO 88:(Recherche d'Information Assistée par Ordinateur). Conference*, pp. 380-388.

Surowiecki, J., 2005. *The wisdom of crowds*. Anchor.

Teevan, J., Liebling, D.J. and Lasecki, W.S., 2014, April. Selfsourcing personal tasks. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems* (pp. 2527-2532). ACM.

Teodoro, R., Ozturk, P., Naaman, M., Mason, W. and Lindqvist, J., 2014, February. The motivations and experiences of the on-demand mobile workforce. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing* (pp. 236-247). ACM.

Treem, J.W. and Leonardi, P.M. (Eds.), 2016a. *Expertise, Communication, and Organizing.* Oxford, UK: Oxford University Press.

Treem, J.W. and Leonardi, P.M., 2016b. What is Expertise? Who is an Expert? Some Definitive Answers. In J. W. Treem and P. M. Leonardi (Eds.). *Expertise, Communication, and Organizing* (pp. 1-24). Oxford, UK: Oxford University Press.

Voss, J.F. and Wiley, J., 2006. Expertise in history. In K. A. Ericsson, N. Charness, P. Feltovich, and R. R. Hoffman, R. R. (Eds.). *Cambridge handbook of expertise and expert performance* (pp. 569-584). Cambridge, UK: Cambridge University Press.

Wexelblat, A. and Maes, P., 1999, May. Footprints: history-rich tools for information foraging. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 270-277). ACM.

White, R. W., Dumais, S. T. and Teevan, J. 2009. Characterizing the influence of domain expertise on web search behavior. *Proc. WSDM '09.* 132-141.

Whittaker, S., Bergman, O. and Clough, P., 2010. Easy on that trigger dad: a study of long term family photo retrieval. *Personal and Ubiquitous Computing*,*14*(1), pp.31-43.

Yakel, E., 2007. Digital curation. *OCLC Systems & Services: International digital library perspectives*, *23*(4), pp.335-340.

Yarosh, Svetlana, Tara Matthews, and Michelle Zhou (2012). Asking the right person: supporting expertise selection in the enterprise. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12). ACM, New York, NY, USA, 2247-2256.

Yi, J., Jin, R., Jain, S. and Jain, A., 2013, March. Inferring users' preferences from crowdsourced pairwise comparisons: A matrix completion approach. In*First AAAI Conference on Human Computation and Crowdsourcing*.

Zhang, J., Ackerman, M. S. and Adamic, L. 2007. Expertise networks in online communities: structure and algorithms. *Proc. WWW '07*. 221-230.

Zhang, H., Law, E., Miller, R., Gajos, K., Parkes, D. and Horvitz, E., 2012, May. Human computation tasks with global constraints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 217-226). ACM.

Zhao, X. and Lindley, S.E., 2014. Curation Through Use: Understanding the Personal Value of Social Media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2431–2440.