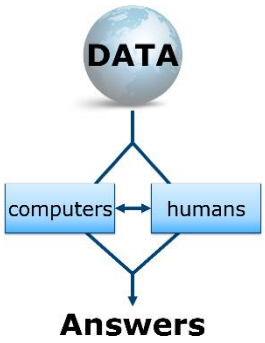
# **Alexander J. Quinn** ▪ Research Statement

Computers have evolved from machines for automating calculations on data to the predominant medium for human communication. Although these two roles may at first seem disjoint, their intersection makes it possible to automate tasks which entail some human work, but are otherwise computational in structure. My research in human-computer interaction is about **solving data-intensive problems by incorporating human contributors into computational processes**. With *human computation,* a user specifies a problem, and then a computer coordinates human workers to perform the steps of the computation that it cannot do well. My dissertation is motivated by two research questions:




**Can a computational process enable people working apart to act as one?**

**How can a messy decision be decomposed into small, coherent human tasks?**

Outside of my dissertation work, I have developed several systems while working on projects led or co-led by professors based in the departments of economics, English, linguistics, and information studies, as well as computer science. Through these collaborations, I have developed a deeply interdisciplinary perspective which is integral to my style of research and future plans.

## Browsing independently—acting as one

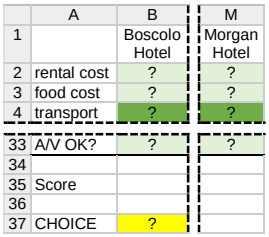
For some decisions, the main obstacle to finding an answer is obtaining all of the relevant data. For example, suppose you were responsible for choosing a site for a meeting of a few hundred people in another city. If you already had a spreadsheet containing all of the relevant alternatives with every attribute that matters to you (e.g., costs, travel times, capacities, etc.), the decision would be easy. Such information is often difficult to compile in an automated way, and yet tedious to gather manually. You could split the work among co-workers with each person gathering every relevant detail about one possible location. To outsource the job, you could post those same slices to a crowdsourcing platform, such as Amazon's Mechanical Turk, where workers will do a few minutes of work for a small amount of money. Either way would be inefficient because they would end up gathering far more information than necessary. If you did it all alone, you would no doubt search for the most important details first and use those to decide what to look at next. However, such an optimization would be difficult to coordinate if the job were split into slices among people working separately.

**AskSheet.** To address the above problem—efficient decision input acquisition—I created AskSheet, a system that uses the structure of the formulas to prioritize the inputs and eliminate those that will not affect the end result. To start, the requester develops a spreadsheet model of the decision with formulas that would calculate the decision result if all of the inputs were already filled in. Cells in need of input are marked by entering a special `=ASK(…)` formula, the parameters to which specify the type of information requested and identify the headings that relate to it. AskSheet uses those parameters together with the data headings and the structure of the model to slice the spreadsheet into small, coherent tasks, each of which entails filling in a few cells of the spreadsheet. It posts the tasks on Mechanical Turk and manages the entire process until a result can be calculated.

| | A | B | M |
|---|---|---|---|
| 1 | | Boscolo Hotel | Morgan Hotel |
| 2 | rental cost | =ASK(..) | =ASK(..) |
| 3 | food cost | =ASK(..) | =ASK(..) |
| 3 | transport | =ASK(..) | =ASK(..) |
| 33 | A/V OK? | =ASK(..) | =ASK(..) |
| 34 | | | |
| 35 | Score | =SUM(.. | =SUM(.. |
| 36 | | | |
| 37 | CHOICE | =INDEX( | |

Skeleton decision model for choosing the site for a big meeting. `=ASK(…)` specifies the inputs.

The key innovation is how it leverages the syntactic structure of all of the formulas in the user's spreadsheet to prioritize the inputs. AskSheet exploits the fact that spreadsheets have no guaranteed order of evaluation. Short-circuit evaluation of operations such as `=AND(…)`, `=OR(…)`, and even `=MAX(…)` can save human effort. To maximize such savings, it continuously prioritizes the inputs by *value of information*, a heuristic that in this case estimates the expected overall savings if that input were acquired next. The effect is as if an individual were researching the alternatives alone and eliminating those that cannot possibly win. In this way, the workers, who do not communicate with each other, are able to *act as one*.



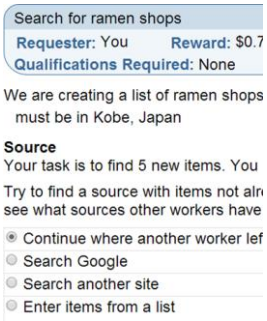
| | A | B | M |
|---|---|---|---|
| 1 | | Boscolo Hotel | Morgan Hotel |
| 2 | rental cost | ? | ? |
| 3 | food cost | ? | ? |
| 4 | transport | ? | ? |
| 33 | A/V OK? | ? | ? |
| 34 | | | |
| 35 | Score | | |
| 36 | | | |
| 37 | CHOICE | ? | |

AskSheet prioritizes the inputs (B2:M33) to save human effort. Dark green cells (B4:M4) are highest priority. The yellow cell (B37) is the final result.

The savings depends heavily on the specific formulas in the user's model. For the personal decisions tested for our paper at CSCW 2014 [1], AskSheet eliminated between 37% and 82% of the inputs. Although that work used Mechanical Turk, the concept and underlying architecture are designed to support other methods of connecting tasks with human contributors, such as email or social network sites.

While AskSheet demonstrated how to efficiently acquire the inputs (fill in the blank cells), a complementary problem is how to enumerate the alternatives (fill in the headings). That is equivalent to using crowd labor to make a list of things, e.g., conference hotels in major cities, restaurants that serve carrot cake, etc. The challenge is that workers do not know which sources and items have been seen.

**Relay.** I recently created a system that enables more efficient list-making by crowds. To start, the requester gives concrete criteria for the kind of list items that are desired. Then, workers think of *sources* where they can find them, e.g., search queries or unfiltered lists found online. Each worker must enter a specified minimum (e.g., ≥5 items per task). If a source has more than that, the worker may either choose to continue for an additional per-item bonus, or *pass the baton* to another worker to finish it (and hence the name *Relay*). For thorough coverage, it tracks which sources have been checked and requires workers to check sources that have not been checked before. Since some duplication is inevitable, Relay uses an advanced autosuggest to help make duplicate list items trivial to detect.

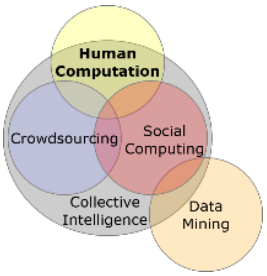



Relay focuses each worker on a single source, and allows them to continue where another left off.

Although Relay is still in progress, I have already tested it on several jobs, making lists of professors researching online education; donut shops in Washington, DC; ramen noodle shops in Kobe, Japan; and articles about food safety.

## More human computation

Before beginning my dissertation work, I compiled a survey and taxonomy of human computation. Having found the literature to be especially segmented, I wanted to help future students and researchers get started. Since its publication at CHI 2011 [5], it has been a required reading in at least a dozen undergraduate and graduate courses at different universities, and has received over 250 citations.



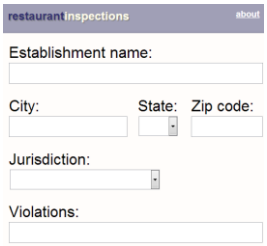

Our survey/taxonomy aims to reconcile several related terms.

**CrowdFlow**. Initially, I set out to study hybrid human computation with machine learning classifiers. I built CrowdFlow, a system that combined the two, making it possible to specify a target quality (accuracy rate), time, or cost [8]. I applied it to problems in computer vision (human detection) and natural language processing (sentiment analysis). The system worked as expected, but the results were not as impressive as we had hoped, so I shifted course.

**CrowdLib**. In the course of building CrowdFlow, I developed a Python toolkit for working with Mechanical Turk. It is at the heart of all of the above systems, as well as another that I built for improving machine translation results [3, 9]. CrowdLib dramatically reduces the amount of code needed to run experiments on the platform. The toolkit is now fully documented and publicly available [4].

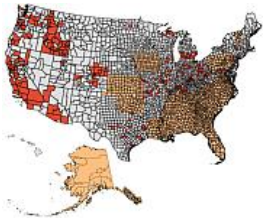
## Unifying local restaurant inspection reports

For the past two years, I have been working with two economics professors and my advisor to build a unified database of restaurant inspection reports scraped from the web sites of local health departments throughout the US. The primary goals are to enable new economics research about the effectiveness and consistency of food safety enforcement, and to support restaurants and regulators in their complementary missions. Since health departments follow no common format for inspections or disclosure, each site requires new scraper code. To bring this up to scale, I built a distributed architecture for scraping the reports. I have supervised four students (3 CS undergraduates and 1 EE/CS master's student) writing modules for new jurisdictions. We now have data for 860,000 restaurants spanning 960 counties in 30 states covering 41% of the US population.



The database covers over 3 million restaurant inspection reports scraped from health department web sites.

This work has enhanced my dissertation, and vice versa. For example, the tedious process we used to list all of the agencies that publish inspection reports online was a major inspiration for Relay. Conversely, some of the internal tools I built for standardizing the data were semi-automatic, inspired by my work with human computation. More broadly, web scraping and crowdsourced data gathering are two sides of the same coin: both facilitate analysis of arbitrary data from the web.


The shaded areas are the counties (red/dark) and states (beige) that post restaurant inspection data. Others are not shown.

## Branching out: Designing for the wild

A core value in the University of Maryland Human-Computer Interaction Lab (HCIL) is that our work should benefit real users. Three of the side projects that supported me as a graduate student have been publicly released and used heavily.

**StoryKit.** As a case study of co-design with children and their grandparents as design partners, I developed a mobile application for story authoring and sharing. Text, drawings, images, and sounds can be positioned on pages, much like in desktop presentation software. Since the release in 2009, StoryKit has been a hit with families and schools alike. In a typical month, kids spend about 100,000 hours creating stories in StoryKit (based on our analysis published in TOCHI [2]).


A child and grandfather create a story in a park.

**Experiencing art history.** The Smithsonian Freer Gallery houses many works so precious that they must often be stored out of sight. Even when on display, visitors may see only a section at a time. Working with curators and the Maryland Institute for Technology in the Humanities, I created applications for a tabletop computer that give visitors a "hands-on" augmented experience with two works: a 14th century Chinese calligraphy scroll [6] and a 15th century painted Persian manuscript [7]. These applications were deployed in the museum for over 2 years.


The applications enable "hands-on" interaction with precious artworks.

**International Children's Digital Library.** As a service to the world's children, my lab maintains a free collection of scanned children's books spanning 61 languages. During my first year, I conducted a lab study comparing three methods of enhancing readability of scanned books. The results (published at CHI [10]) led to enhancements to the site, which serves about 100,000 unique visitors per month.


*PopoutText* preserves artistic typographical features while improving readability.

## Research agenda

Behind the inherent novelty of the "remote person call" is the ongoing transformation to a more fluid labor market. I aim to discover structures and methods that will expand the range and scale of data-intensive tasks that can be accomplished using efficient pairing of computers and human contributors.

**Amplify workers' efforts.** Most crowd work today is repetitive and tedious. However, a promising pattern is exemplified by the tools used by professional translators, which leverage machine automation wherever possible, but rely on a human expert for critical judgments. I will work with researchers in areas such as computer vision and entity resolution to try similar approaches to their problems.

**Support data analytics.** As I have learned by working with the food safety data, our ability to fully utilize the troves of available data is constrained by the inability of current platforms to standardize and interpret it. Data mining is an extremely useful tool, especially for very large datasets, but where a human eye is required, we must invent ways to adapt the work to this new kind of workforce. I will join forces with data mining researchers to explore models that combine data mining with human computation to extract more value from large data sets.

**Design for an equitable marketplace.** Technologies that violate social expectations tend to be short-lived. Therefore, these systems will only succeed if the underlying economic and ethical issues can be addressed. As technologists, it is up to us—not policy makers or social scientists—to design with consideration for issues such as trust, accountability, and fairness. Partnering with researchers in business and labor economics, I will look for ways to guide the development of these platforms in sustainable directions. In particular, I hope to discover if a platform might encourage fair pay by analyzing and communicating information about the match—or mismatch—of the compensation to the amount of work.

## References


1. **Quinn, A.J.**, Bederson, B.B. 2014. AskSheet: Efficient Human Computation for Decision Making with Spreadsheets. In *Proc. of ACM Conf. on Computer Supported Cooperative Work (CSCW '14)*. 11 pages. *In press.*
2. Bonsignore, E., **Quinn, A.J.**, Druin, A., Bederson, B.B. 2013. Sharing stories in "in the wild": A mobile storytelling case study using StoryKit. In *ACM Transactions on Computer-Human Interaction (TOCHI)*. 20:3:18. 38 pages.
3. Resnik, P., Buzek, O., Kronrod, Y., Hu, C., **Quinn, A.J.**, Bederson, B.B. 2013. Using Targeted Paraphrasing and Monolingual Crowdsourcing to Improve Translation. In *ACM Transactions on Intelligent Systems and Technology (TIST)*. 4:3:38 (July 2013). ACM. 21 pages.
4. **Quinn, A.J.** 2013. CrowdLib documentation. http://www.cs.umd.edu/hcil/crowdlib
5. **Quinn, A.J.**, Bederson, B.B. 2011. Human computation: a survey and taxonomy of a growing field. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI '11)*. ACM. 10 pages.
6. **Quinn, A.J.**, Kvernen, E., Lester, D., Fraistat, N. 2011. A Breath of Spring. http://vimeo.com/25439034
7. **Quinn, A.J.**, Kvernen, E., Lester, D., Fraistat, N. 2011. Haft Arang. http://vimeo.com/25438734
8. **Quinn, A.J.**, Bederson, B.B., Yeh, T., Lin, J. 2010. CrowdFlow: Integrating Machine Learning with Mechanical Turk for Speed-Cost-Quality Flexibility. Technical Report HCIL-2010-09, University of Maryland.
9. Resnik, P., Buzek, O., Hu, C., Kronrod, Y., **Quinn, A.**, Bederson, B.B. 2010. Improving translation via targeted paraphrasing. In *Proc of Empirical Methods in Natural Language Processing (EMNLP '10)*. ACL. 11 pages.
10. **Quinn, A.J.**, Hu, C., Arisaka, T., Rose, A., Bederson, B.B. 2008. Readability of scanned books in digital libraries. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI '08)*. ACM. 10 pages.