Abstract of “Public Data Systems for Evolving Information”, by Shaun Wallace, Ph.D., Brown University, October 2023.

The data around us evolves rapidly. Continuous human effort is the life support that maintains its quality. A new type of public system, combining crowdsourcing and peer production, can enable like-minded users to engage with and maintain evolving data.

I developed two public data systems deployed longitudinally in the wild: Drafty and Sketchy. Drafty is a tabular dataset of Computer Science professors that has evolved annually for eight years. During four-minute tasks, Sketchy's users simultaneously contribute sketches and view their peers' contributions to maintain inspirational stimuli.

This dissertation shows that unpaid everyday visitors make higher quality contributions when they have more domain-specific knowledge, especially compared to paid crowdworkers. Drafty automatically extracts user interest from their passive interactions to prompt and elicit accurate contributions. However, more is needed to increase the likelihood that someone contributes. Sketchy shows that providing users autonomy over their interactions to access randomized examples from its evolving dataset motivates higher-quality contributions. Integrating these lessons, Drafty is re-developed to give users similar freedom and functionality. Drafty generates dynamically generated insights (Databaits) from its evolving data. A naturalistic study shows that, unlike users in traditional recommendation systems, Drafty's users prefer randomized insights. Like Sketchy, when Drafty's users interact with these insights from its source data, they make more accurate contributions.

What motivates paid and unpaid users to contribute to public data systems? This dissertation develops a theoretical framework that explores the trade-offs people make when selecting different crowd contribution tasks. While money is the strongest motivator for paid crowdworkers, unpaid everyday users are intrinsically motivated by their perception of tasks (i.e., their interest and if their contribution might help others). All users prefer shorter, easier tasks where they contribute their specialized knowledge. Evidence shows that unpaid everyday users make trade-offs, preferring longer tasks that pay less if those tasks are interesting and their contributions can help others. Highly accurate users make these same trade-offs. Together these public data systems demonstrate longitudinal evidence of engaging anonymous users in the wild to view and maintain quality evolving data.

Public Data Systems for Evolving Information

By

Shaun Wallace

B.S., University of Rhode Island, 2008

M.S., University of Limerick, 2009

M.S., Brown University, 2017

Thesis

Submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in the

Department of Computer Science at Brown University

PROVIDENCE, RHODE ISLAND

October 2023

This dissertation by Shaun Wallace is accepted in its present form by the Department of Computer Science as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date 07/12/2023 | 6:36 PM EDT

DocuSigned by:
Jeff Huang
53CA2F8E3CD1405...

Jeff Huang, Advisor

Recommended to the Graduate Council

Date 07/14/2023 | 5:57 PM EDT

DocuSigned by:
John F. Hughes
CD7758DDF7A345C...

John F. Hughes, Reader

Date 07/17/2023 | 7:12 AM EDT

DocuSigned by:
481347ED040A47F...

Zoya Bylinskii, Reader

(Adobe Research)

Approved by the Graduate Council

Date ____________

Thomas A. Lewis, Dean of the Graduate School

Curriculum Vitae

Shaun Wallace has lived in the United States (Indiana, Tennessee, Texas, and Rhode Island) and Ireland (Limerick). He received his B.S. in Management Science and Information Systems from the College of Business at the University of Rhode Island in 2008. He received his M.S. in Music Technology from the College of Computer Science and Information Systems at Limerick University in Ireland in 2009. He joined Brown University Computer Science as a Masters student, taking his first class in Fall 2014. At Brown, he was awarded his M.S. in 2017 and completed his Ph.D. in Computer Science under the advisement of professor Jeff Huang in 2023. As a Ph.D. student, he worked full-time as the Lead Systems Programmer for the department of Computer Science at Brown University. He also completed a research internship at the Adobe Creative Intelligence Lab in 2019, where his research eventually led to the creation of the Readability Consortium. In Fall 2023, he will begin a new position at his first alma mater, the University of Rhode Island, as an Assistant Professor of Computer Science.

Acknowledgments

I especially want to thank my wife and our two children for going with me on this journey that started nine years ago in the summer of 2014. A PhD can be a burden, especially with a young family, and my wife has been incredibly honest and supportive throughout all the milestones and the ups and downs. Love you. :)

To my advisor Jeff Huang, for giving me a chance to do research as a master's student and then helping to create a unique PhD environment where I could pursue research while caring for my family. Thank you for challenging me and always pushing me to be my best and improve. Jeff is the most timely and dependable person I have worked with. The absolute opposite of the absent advisor. Thank you for the guidance, perspective, and journey. I am forever grateful to Jeff, Shriram Krishnamurthi, Ugur Çetintemel, John Bazik, and the Computer Science department for creating and supporting my unique PhD environment. I was incredibly lucky to work with Jeff and such an amazing department.

To my committee members, John F. Hughes and Zoya Bylinskii, I have immeasurably benefited from your advice and perspective for this dissertation and my career. You helped craft my message and improve the validity of my results. You have made me a better researcher, teacher, and scientist. Thank you for the many great conversations and perspectives.

This dissertation would not have been possible without the help and guidance of so many amazing people I have been fortunate enough to collaborate with: Talie Massachi, Alexandra Papoutsaki, Luis Leiva, Hua Guo, Ben D Sawyer, David B Miller, Malte Schwarzkopf, Nediyana Daskalova, Jing Qian, Tongyu Zhou, Zainab Iftikhar, Ji Won Chung, Brandon Woodard, Jiaqi Su, Long Do, Diana Lee, Zhengyi Peng, Matthew Bejtlich, Brendan Le, Neilly Tan, Michael Bardakji, Ari Kintisch, Andrew Park, Aman Haq, Aarthi Anbalagan, Linda Chang, Gabrielle Bufrem, Abraham Peterkin, Yirui Huang, Lucy Van Kleunen, Marianne Aubin-Le Quere, Elijah Rivera, Ugur Çetintemel, Eli Upfal, Ellie Pavlick, Mathew Borton, Jill Pipher, and Björn Sandstede.

Beyond my family and collaborators, I was lucky to have several people who acted as a core support group, from my co-workers at Brown's Computer Science department, ICERM, and Adobe. In particular, Kathleen Billings, Paul Vars, and Talie Massachi. I honestly can never thank you for the countless hours of listening and sharing stories. I would not have made it through without you. To everyone I have mentioned and those I have not, thank you for helping to make my dream come true.

This dissertation was supported by the National Science Foundation under Grant No. IIS-1552663.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

What is a Public Data System

This dissertation establishes and explores the idea of a crowd powered public data system. A public data system is an anonymous collaborative environment where anyone can freely interact and contribute information. The systems pre-define how to organize and display information. It combines aspects of crowdsourcing and peer production. As in crowdsourcing, the interactions and ways to contribute are pre-specified and controlled by the system. However, like peer production, users possess diverse motivations and can freely choose to visit, explore, engage with, and contribute information within the pre-specifications of the system.

Thesis Statement

Most public data systems require continual data maintenance to keep the information up to date and relevant. Creating and maintaining an accurate public dataset can attract everyday users with the interest and domain-specific knowledge required to make accurate contributions. These everyday unpaid users make more accurate contributions than paid users. In a public data system, everyday users prefer to see insights and examples generated from the data. Providing users the right incentives to engage with meaningful insights from accurate data motivates them to continuously interact with and contribute data to create a cycle of perpetual upkeep. A theoretical framework describes user motivation and data decay to explore how incentives and temporal factors affect the potential correctness of evolving information over time.

This dissertation establishes these claims by describing the continued development of two example public data systems, Drafty and Sketchy, and several associated studies observing the real-world behavior of their users over eight years.

1.1 Motivation

The data around us is dying; it starts to lose its relevance almost the moment it is created. Some data evolves over time due to temporal and external factors. For example, university closures and mask requirements during the COVID-19 pandemic changed rapidly, whereas the course a university offered in Fall 2019 will not change [3]. While researchers have studied how to quickly and cheaply collect accurate data [173], the life cycle of this data relies on maintenance to ensure its accuracy over time. Many human-powered crowdsourcing and peer production efforts to collect data, such as Wikis, Forums, free and open-source software, and Knowledge Graphs. Evolving data can have life cycles where it is collected, matures, eventually dies, and becomes obsolete [96]. The maturation process is defined by a continual effort to improve the overall correctness and quality of the dataset. If data decays (i.e., changes) too fast, this can shorten the length of the maturation phase. To delay a dataset's death, people need the motivation to make enough edits to overcome the rate data decays and evolves. Hence, if an evolving dataset's decay rate is high, it will require many accurate contributions. A public dataset allows anyone with enough motivation to contribute to the data and provides transparent ownership of these contributions. A core issue with any public dataset is attracting and maintaining a community of users to contribute to and maintain data over time.

Examples of data decay and the potential death of specific datasets are documented on popular platforms like Wikipedia [59, 156]. For example, Wikipedia's policies to manage its rapidly growing community have made it difficult to recruit enough new editors to replace the previous editors who are leaving [88]. There is even a Wikipedia project[1] dedicated to recording editors with more than 1000 edits who left. To make matters worse, the Wikipedia editors leaving can possess the domain-specific knowledge that paid crowdworkers lack [59]. Wikipedia editors leave because of strict policies or toxic behaviors, while other reasons are challenging to uncover [59]. Wikipedia and WikiData regularly rely on a few users making most edits per article or topic area [181]. Thus, when these users stop contributing, this can cause the data to decay quickly. When evolving data becomes inaccurate, this decreases its benefit to society and can lead to dire consequences [172]. Thus, to maintain the accuracy of its evolving information, a public data system must engage interested users over time.

Two alternative methods to update evolving data that would fail to scale over time are automation and paying crowdworkers. Automated methods, such as web scraping, could scrape and monitor public information on the web. However, this is difficult to scale because the design of web pages is too inconsistent and changes over time [42]. It can be simpler to rely on the efforts of a community of users to update the data. The functionality of web scraping programs would have to be perpetually updated to scrape accurate data. Another alternative is paying crowdworkers, which presents a simple extrinsic motivator to update data quickly. However, it can be challenging to find crowdworkers with the required domain-specific knowledge for some datasets [86]. Also, while paying crowdworkers can improve the

[1] https://en.wikipedia.org/wiki/Wikipedia:Missing_Wikipedians

accuracy of data during the short term, this becomes impractical over time due to the financial constraints of constantly reviewing and updating evolving data [152]. Previous longitudinal research, such as precision crowdsourcing, has focused on the numbers of contributions compared to their accuracy [242]. This dissertation explores how a public data system can rely on motivating a community of users with a mutual interest in the contributed data to provide domain-specific knowledge and accurate contributions over time, see Figure 1.1.

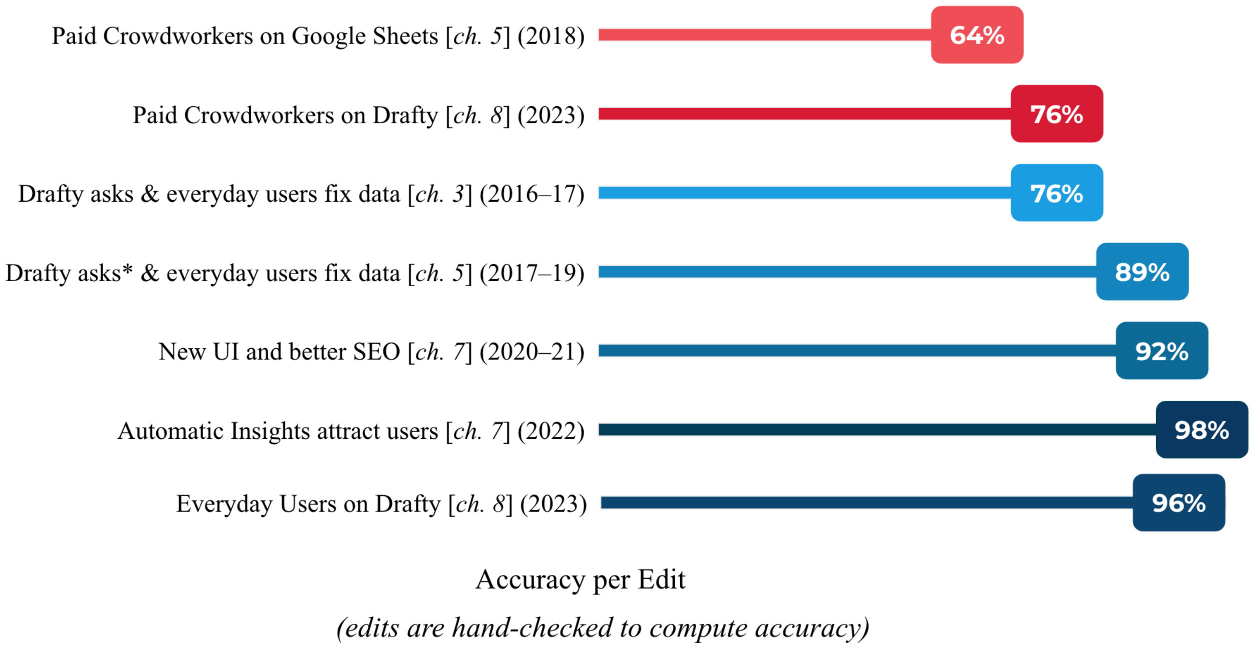


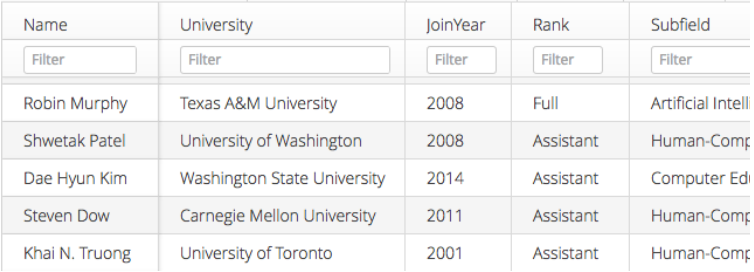
Figure 1.1: The accuracy per edit from the studies in Chapters 4, 6, and 8. Over time, as Drafty integrates new features and methods to motivate contributions, the accuracy per edit increases. The accuracy from Chapter 4 includes everyday users making edits and Drafty asking everyday users to fix data matching and not matching their interests. Whereas, in Chapter 6 Drafty never asks people to fix data not matching their interests.

1.2 A Theoretical Framework to Describe Maintaining Evolving Information

This section covers three important aspects of this dissertation that inform why information evolves and how users' contributions can maintain evolving information. 1) Tabular data is the most common data structure studied in this dissertation. 2) Why does information evolve, and how does evolving information differ from static information? 3) A theoretical framework describing how user motivation and data decay contribute to the overall correctness of evolving information over time.

1.2.1 Structure of Tabular Datasets

The majority of the dissertation studies people editing tabular data presented in spreadsheets. In spreadsheets, tabular datasets are organized in tables with horizontal rows and vertical columns [143, 221]. Each row can represent a unique entity with the same number of cells. Each cell corresponds to the intersection of a specific row and column. Columns correspond to a particular property characterizing data per row. Tabular datasets exist in various formats (e.g., comma separated values or tab separated values) and can be edited using dedicated spreadsheet programs like Microsoft Excel or Google Sheets. This structure means that sucha table constitutes one or more "relations" between the data in each column. Most often, the first column is an index variable for these relations (i.e,. appears only once per table). The majority of the chapters in this dissertation focus on enabling groups of people to collaboratively contribute to high-quality relational data, also known as "relational tables" [239]. For example, chapters 3, 5, 7, and 8 focus on the maturity phase of a tabular dataset and extend prior work on tabular data from Wikipedia and Wikidata [26, 72]. The tabular dataset used in these chapters focuses on a semantically cohesive concept, Computer Science professors (Figure 1.2), and spans thousands of rows and, at times, more than ten columns.



Name	University	JoinYear	Rank	Subfield
Filter	Filter	Filter	Filter	Filter
Robin Murphy	Texas A&M University	2008	Full	Artificial Intell
Shwetak Patel	University of Washington	2008	Assistant	Human-Comp
Dae Hyun Kim	Washington State University	2014	Assistant	Computer Ed
Steven Dow	Carnegie Mellon University	2011	Assistant	Human-Comp
Khai N. Truong	University of Toronto	2001	Assistant	Human-Comp

Figure 1.2: A partial and unordered view of Drafty's initial spreadsheet interface used in chapters 3 and 5. Different profiles of computer science faculty and their professional development can be seen.

1.2.2 How and Why do Public Data Systems' Information Evolve?

Extending the life of data, i.e., the time during which it remains valid, requires perpetual human attention. External temporal factors cause data to decay, thus, destroying its utility. Drafty is a public data system developed throughout this dissertation that hosts a publicly editable tabular dataset with thousands of academic profiles of Computer Science faculty from the United States and Canada. Each academic profile is a row in a spreadsheet. The columns correspond to a professor's affiliated university, the year they joined as faculty, subfield area of expertise, and where they received their Bachelor's and Doctorate degrees. This data evolves, or decays, annually based on new faculty hires and annual conferences and publications that indicate a professor is shifting their research area.

There is some information in a Computer Science professor's academic profile that is not static: their full name, the university that employs them, the year they joined their current university, and their subfield area of expertise (i.e., primary research area). The external factors that cause the data to change include a professor moving universities, a new hire at a university, a professor retiring, a professor leaving to go to an industry position, or changing their subfield area of expertise. Thus, these external factors cause the data to change annually due to tenure appointments or hiring season or slower if a professor changes their primary research area. On the contrary, other data types are considered static. For example, it is unlikely that a tenure-track faculty would receive a new Bachelors's or Doctorate related to Computer Science.

In contrast, chapter 6 explores a form of rapidly evolving information, sketches created during 4-minute sketching tasks within the public data system Sketchy. Sketchy automatically records and synchronizes each sketch as a series of XY points and colors per stroke. Users can view each rapidly evolving sketch in virtual rooms using Sketchy's Peek feature. The quick creation process propels users to rapidly create and seek related sketches to complete the sketching task. This rapid co-creation of evolving sketches enables groups of people to contribute to evolving information. This action overcomes the static limitations of prior research, where people would have to collect numerous static sketching examples before running a sketching activity. Even static sketches will decay in their usefulness over time. For example, sketches from decades ago might only be limited to desktops, not mobile devices. Or the subjectivity of static images might change. For instance, how icons map to different functionalities will gradually shift for different age groups [4]. Sketchy provides a mechanism to quickly collect and share evolving sketches as inspirational stimuli to users in real-time to overcome these temporal factors.

While this dissertation mainly focuses on Drafty's evolving data, which changes annually, evolving data can change at different rates, like with Sketchy. A real-world example of evolving data outside this dissertation is a COVID-19 dataset of University policies [3] that changed almost weekly as schools responded to federal policies and outbreaks. This data decays rapidly. Thus, the community contributing to the data requires enough motivation to overcome this decay. The pandemic provided the motivation stimulus for the community to collect and share this evolving data. This dissertation focuses mainly on Computer Science faculty profiles because it provides enough time to develop analysis and new features within a public data system to continually motivate its users to contribute.

1.2.3 Describing the Theoretical Framework to Maintain Evolving Information

When building and maintaining evolving data, there is the current dataset users modify and the ideal ground truth version of it that changes over time. If the current dataset and the ground truth are the same, then the current dataset is 100% correct. While creating a 100% correct dataset is attainable if the data is static, correctness will change over time as data evolves. The data is evolving because the ground truth is changing over time. Over the period data evolves, users can modify the current dataset to maintain its correctness. The current dataset is D, and the ground truth is G. See Figure 1.3 for an example.

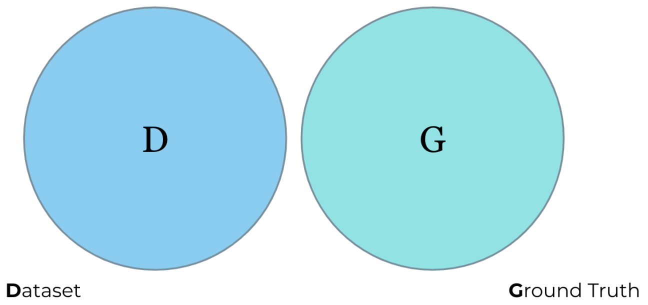


Figure 1.3: The current dataset is D. Users can freely modify it. The ground truth version of the current dataset is G. The ground truth is ephemeral; it evolves due to real-world events and requires continuous maintenance to ensure it is included in the current dataset.

This theoretical framework describes how user contributions and natural changes to the ground truth affect the correctness of an evolving dataset over time. It defines how different user contributions to the current dataset affect correctness. It also describes the coefficients and units that affect user motivations to make contributions and their accuracy. This framework lays the groundwork to understand what types of systems, tasks, and features someone could develop to motivate users to overcome how the ground truth's data evolves and decays over time.

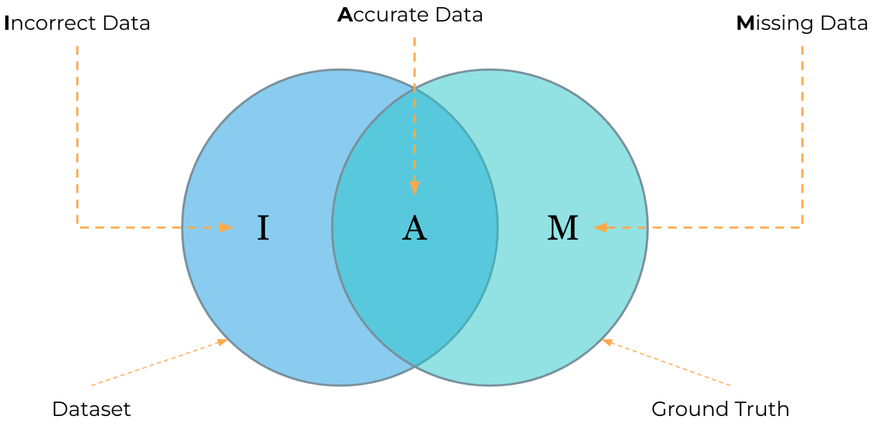


Figure 1.4: Incorrect data (**I**), accurate data (**A**), and missing data (**M**) are derived from the dataset (**D**) and the ground truth (**G**).

The first concept to describe is how data points within an evolving dataset and its ground truth overlap to determine incorrect, accurate, and missing data. See Figure 1.4 for a visual explanation. Inaccurate data (**I**) is in the current dataset but not the ground truth; see Equation 1.1. Inaccurate data is the difference between **D** and **G**. Inaccurate data must be removed from the current dataset to improve correctness by deleting or correcting it.

$$\mathbf{I} = \mathbf{D} - \mathbf{G}. \tag{1.1}$$

Accurate data (**A**) is in the current dataset and the ground truth; see Equation 1.2. Accurate data (**A**) is the intersection of **D** and **G**.

$$\mathbf{A} = \mathbf{D} \cap \mathbf{G} \tag{1.2}$$

Missing data (**M**) is not in the current dataset but is in the ground truth; see Equation 1.3. Missing data is the difference between **G** and **D**. Missing data needs to be added to the current dataset to improve correctness. Missing data from the ground truth can be difficult to predict, especially for evolving data, as it is ephemeral.

$$\mathbf{M} = \mathbf{G} - \mathbf{D} \tag{1.3}$$

A dataset's correctness (C) is the number of accurate data points divided by the number of inaccurate, accurate, and missing data points; see Equation 1.4. If there is no inaccurate or missing data, the dataset is 100% correct. It contains no incorrect data and is not missing anything. Predicting the correctness of an evolving dataset is complex because the amount of missing data can change without anyone knowing.

$$C = \frac{\mathbf{A}}{\mathbf{I} + \mathbf{A} + \mathbf{M}} \tag{1.4}$$

To improve a dataset's correctness (C), removing incorrect data has the same effect on correctness as adding missing data to the current dataset. However, there are multiple ways users modify a dataset and how data points will change over time. A key concept is how users and real-world events modify incorrect data (**I**), accurate data (**A**), and missing data (**M**) over time in different ways. See Table 1.1 for an overview of how data changes and how user contributions to the current dataset and real-world events evolve the ground truth to affect correctness.

Predicting or assessing what data is missing (**M**) can be time consuming and is an open research question across different types of data [146, 246]. Comparing the incorrect and accurate data from the current dataset is more feasible to assess the quality or accuracy of an evolving dataset. In this framework, accuracy (α) builds on the idea of the precision metric from the information retrieval community; see Equation 1.5.

$$\alpha = \text{Accuracy} = \frac{\mathbf{A}}{\mathbf{I} + \mathbf{A}} \tag{1.5}$$

To simplify things, let us suppose that the set **A** contains n items and look at what can happen when α starts at 0.5. In this situation, we must have the denominator, $\mathbf{I} + n$, be twice the numerator, so **I** must be n as well, and we have Equation 1.6.

$$\alpha_{init} = \frac{n}{n+n}. \tag{1.6}$$

Now suppose that users contribute k more items, and half of these are accurate. Then both **A** and **I** increase by $k/2$,

		How does the contribution modify the size of data (↑↓) and affect correctness (color)?		
User Contribution	Is it Correct?	**I**ncorrect Data	**A**ccurate Data	**M**issing Data
Add New Data	Yes		🟩 ↑	🟩 ↓
Add New Data	No	🟥 ↑		
Delete Existing Data	Yes	🟩 ↓		
Delete Existing Data	No		🟥 ↓	🟥 ↑
Modify Existing Data	Yes	🟩 ↓	🟩 ↑	🟩 ↓
Modify Existing Data	No	🟥 ↑	🟥 ↓	🟥 ↑

		How does data evolving modify the size of data (↑↓) and affect correctness (color)?		
Data Evolves	In Current Dataset?	**I**ncorrect Data	**A**ccurate Data	**M**issing Data
Add New Ground Truth Data	No			🟥 ↑
Add New Ground Truth Data	Yes	🟩 ↓	🟩 ↑	
Remove Ground Truth Data	No			🟩 ↓
Remove Ground Truth Data	Yes	🟥 ↑	🟥 ↓	

Table 1.1: The top half of this table shows how different types of user contributions affect incorrect data (**I**), accurate data (**A**), and missing data (**M**). "Is it Correct" refers to the data point the user contributed. The bottom half of this table shows how different ways data evolves affect incorrect data (**I**), incorrect data (**A**), and missing data (**M**). (**A**), and missing data (**M**). "In Current Dataset" asks if the ground truth data point is in the current dataset. A green background indicates a change that will increase the correctness of an evolving dataset. A red background indicates a change that will decrease the correctness of an evolving dataset. Notably, when a user modifies existing data, they simultaneously delete one piece of data and add another. Thus when a modification is correct, the arrows in the table match, adding new data and deleting existing data.

so the denominator increases by k, and we get Equation 1.7 as well.

$$\alpha_{final} = \frac{n+k/2}{(n+k/2)+(n+k/2)} = \frac{n+k/2}{2(n+k/2)} = 0.5 \tag{1.7}$$

In fact, regardless of the accuracy of the new contributions, the denominator will always increase by k. If the numerator increases by more than $k/2$, then α_{final} will be greater than 0.5; if it increases by less than $k/2$, then α_{final} will be less than 0.5. In an extreme case, suppose that someone deletes all the accurate data so that $k = -n$. Then after this edit, **A** will contain no items, but **I** will still contain n, and α will be $0/(n+0) = 0$.

If we have an existing dataset, the accuracy of contributions that only modify existing data needs to be greater than 50%. This will increase the accuracy of the dataset over time until all incorrect data becomes accurate. This describes the typical scenario of people editing an existing dataset. However, an evolving dataset is constantly growing and shrinking as data evolves. It is possible over time to find a subset of the accurate data by manually checking if contributions are correct or incorrect. Relying on automated methods, such as web scraping, could be inconsistent, especially for subjective data that requires domain-specific knowledge. For example, Chapters 4 and 5 show that paid crowdworkers often believe the research areas of Machine Learning and Computing Education are similar. The paid crowdworkers from these chapters most likely mistakenly believe the terms "learning" and "education" are the same.

1.2.4 Increasing User Motivation to Maintain Evolving Information

To maintain evolving data, whose accuracy at time t is α_t, over a time period of duration s, i.e., during the period of time ($[t, t+s]$), the proportion of accurate data points added among all data added must be at least α_t (Equation 1.8) Similarly, if the correctness at time t is C_t, then to maintain that level of correctness, the quotient of accurate data over all data either (a) added to the dataset or (b) new missing data must again be C_t (Equation 1.9).

To improve accuracy and correctness, accurate contributions must overcome people's mistakes and newly evolved data or changes in existing data.

$$\frac{A_{t+s}}{I_{t+s} + A_{t+s}} \geq \frac{A_t}{I_t + A_t} \tag{1.8}$$

$$\frac{A_{t+s}}{I_{t+s} + A_{t+s} + M_{t+s}} \geq \frac{A_t}{I_t + A_t + M_t} \tag{1.9}$$

To evaluate an evolving dataset over time, it is recommended to keep the duration s of the measurement interval constant. For example, does data evolve due to real-world events that repeat yearly? If yes, then s could be one year. For example, Chapter 5 shows the faculty job market and Ph.D. application seasons affect Drafty's data annually. Developing domain-specific knowledge on why data evolves will ensure singular noisy events will not provide an incorrect interpretation of a public data system's community of contributors. For example, if s were one day, then a single user vandalizing the dataset would greatly affect how we interpret the evolving dataset's correctness compared to the contributions over one year.

Evolving data is ephemeral and will change over time. This framework describes how user motivation can increase contributions to overcome these natural changes in data. See Figure 1.5 for a visual example of how user contributions can overcome changes in the ground truth.

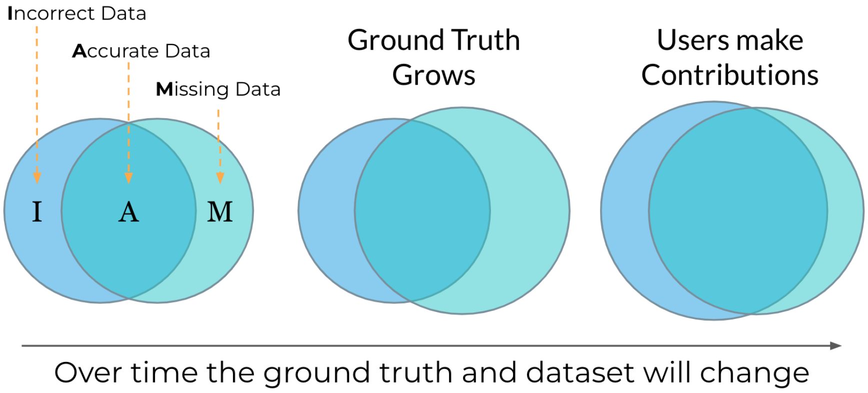


Figure 1.5: If data evolves, users must contribute an equal or greater number of accurate data points to maintain the current dataset's correctness.

The correctness of an evolving dataset will converge over time as t approaches infinity. A thought experiment to examine this long-term behavior is each year pick a number of contributions uniformly between some min and max value. Compute (**I**), (**A**), and (**M**) per year. Then iterate until correctness converges. This allows us to look at the expected values ($\mathbb{E}$) for inaccurate (**I**), accurate (**A**), and missing (**M**) data at any given time (t). We include time (t) to update our expectations of **I**, **A**, and **M** as we learn more over time about how these grow and shift. Hypothetically, when studying changes for accurate data over time, once t is at infinity, we will learn the upper and lower bounds and the true $\mathbb{E}(\mathbf{A})$. The upper and lower bounds are the minimum and maximum new data points added to a set at a given period of time t divided by two; Equation 1.10. The minimum and maximum new data points are uniformly sampled.

$$\mathbb{E}(\mathbf{X}) = \frac{\min_{\forall \mathbf{X}} + \max_{\forall \mathbf{X}}}{2} \tag{1.10}$$

The same thought process for $\mathbb{E}(\mathbf{A})$ applies to $\mathbb{E}(\mathbf{I})$ and $\mathbb{E}(\mathbf{M})$. Until that point, we can only estimate based on our current knowledge at time t. Therefore, based on the expected values over time, as time (t) approaches infinity, we can understand where correctness (C) converges; see Equation 1.11.

$$\lim_{t\to\infty} C_t = \lim_{t\to\infty} \frac{A_t}{I_t + A_t + M_t} = \lim_{t\to\infty} \sum_{i=1}^{t} \frac{A_i}{I_i + A_i + M_i} = \frac{\mathbb{E}(\mathbf{A})}{\mathbb{E}(\mathbf{I}) + \mathbb{E}(\mathbf{A}) + \mathbb{E}(\mathbf{M})} \tag{1.11}$$

Since $\mathbb{E}(\mathbf{A})$ represents the expected number of accurate contributions, we can see that $\mathbb{E}(\mathbf{A}) = \mathbb{E}(\mathbf{D}) \cdot \alpha$, where α is the accuracy of contributions from Equation 1.5 at any given time (t). $\mathbb{E}(\mathbf{D})$ is the number of expected new contributions from users. Similarly, $\mathbb{E}(\mathbf{I}) = \mathbb{E}(\mathbf{D}) \cdot (1-\alpha)$. By making these substitutions, we can modify Equation 1.11 to see the impact of accuracy on where correctness converges; see Equation 1.12.

$$\lim_{t\to\infty} C_t = \frac{\mathbb{E}(\mathbf{D}) \cdot \alpha}{\mathbb{E}(\mathbf{D}) + \mathbb{E}(\mathbf{M})} \tag{1.12}$$

What makes an evolving dataset more difficult to maintain compared to a static dataset is that the missing data (**M**) will naturally change over time as the ground truth evolves. The greater the expected values for new evolving data ($\mathbb{E}(M)$) compared to the expected values of the number of new data points users contribute ($\mathbb{E}(\mathbf{D})$), the faster data evolves. The presence of evolving data (**M**) will ensure where correctness (C) converges to will always be less than the accuracy (α) of user contributions. This is why relying on accuracy (α) alone to evaluate an evolving dataset can be misleading. Accuracy (α) could be high if everyone is only correcting incorrect existing data. In this scenario, the missing data (**M**) will keep growing, thus lowering correctness over time. As previously mentioned predicting missing data (**M**) over time is difficult. That is why motivating a high number of accurate contributions to overcome people's mistakes and missing data is essential to maintaining an evolving dataset.

To maintain data, it is imperative to understand how to motivate people to contribute data and what types of people make accurate contributions. To ensure the correctness of an evolving dataset converges towards a higher number, we need to increase accurate contributions and decrease inaccurate contributions while ensuring there are enough contributions to overcome how quickly data evolves.

The most common method to motivate contributions is rewarding monetary compensation for contributing. Prior efforts often use monetary compensation as an extrinsic motivator to quickly motivate people to contribute data [122]. However, developing methods to appeal to a user's intrinsic motivation, such as showing them their contribution will help others, can also increase the likelihood they will contribute data [160, 200]. The amount of effort required to contribute, or a person's level of interest, can also be coefficients that modify the likelihood someone will contribute. Each of these methods can be viewed as external factors affecting user motivation and the accuracy of their contributions.

User motivation is the likelihood a user contributes data within one visit. User motivation is similar to the conversion rate metric to determine if a user will perform a specific action during a period of time. If enough people contribute accurate data, that should overcome the number of incorrect contributions and natural changes over time. Converting visitors who do not contribute (i.e., lurkers) into contributors is a complex scenario within crowd-powered and peer production systems [18].

User motivation is a function of a number of normalized external factors; see Equation 1.13. Each external factor (m) will modify the likelihood that someone contributes (i.e., conversion rate).

$$\prod_{i=1}^{n} \frac{m_i}{|m|} \tag{1.13}$$

Similar to user motivation, the quality of contribution (i.e., accuracy) is also a function of a number of normalized external factors; see Equation 1.14. Each external factor (s) will modify the likelihood someone makes an accurate contribution.

$$\prod_{i=1}^{n} \frac{s_i}{|s|} \tag{1.14}$$

These external factors can describe a task to contribute (money, effort, user interest, time to complete task), features in a public data system (recommendations, design, data ownership), or the people themselves (how they are recruited, data ownership). They apply to paid and unpaid scenarios in crowdsourcing and peer production environments. People design tasks and systems around these external factors. Understanding these external factors makes it possible to translate findings from crowdsourcing to peer production and back again.

Each external factor can affect the other. For example, if the amount of money to contribute is increased, the less interested a user needs to be in the data related to the task. If there is no money, then the user's interest in the data must be higher to increase the conversion rate and elicit accurate contributions. Within crowdsourcing or peer production, researchers can create a mental model for how their task designs and systems might affect user motivation and the accuracy of user contributions.

This dissertation explores these external actors across different groups of people in multiple public data systems.

By focusing on tasks, features, and methods to implement the coefficients to increase motivation, the likelihood a visitor contributes and their number of contributions should increase. Some examples of coefficients explored in this dissertation are new features generating randomized automatic insights within the system or developing dynamically updated analyses of the system's data to increase user motivation. This dissertation also explores how the existence and preference of these coefficients vary across different types of people. For example, unpaid everyday visitors to a public data system who want to help their community compared to paid crowdworkers whose primary motivator (i.e., coefficient) is likely payment per task. Thus, by studying how user motivation affects the number of contributions and the average accuracy of different groups of people's contributions, this dissertation shows how to build public data systems to attract and motivate users to make accurate contributions over time to maintain evolving data.

1.3 Accuracy and Visitors Needed to Maintain an Evolving Dataset

For someone building a public data system to maintain evolving information, knowing the constraints and bounds of visitors' editing behaviors and how many data points can evolve will help them understand if their system's visitors can maintain its evolving data over time. Over a period of time, data can evolve in two ways. First, data in the current dataset (D) can become outdated and require changes. Equation 1.15 uses the probability a data point in the current dataset (D) will become outdated over a given period of time (e) to describe the number of evolving data points from the current dataset (D) that require user contributions to correct.

$$e \cdot D \tag{1.15}$$

Second, the ground truth (G) can grow, and the number of new data points that need to be added is defined as (U). The Number of Evolved Data Points, Equation 1.16, combines the two ways data can evolve, to define the number of data points that have evolved and those that require user contributions to correct.

$$(e \cdot D) + U \tag{1.16}$$

It is important to separate $(e \cdot D)$ and U because datasets can evolve differently. Take a tabular dataset of academic faculty job postings. While new rows of data need to be added frequently, the individual values per row (i.e., deadline, university name, etc.) would rarely change. While in Drafty's computer science professor profiles page, existing data will change when professors move to a different university or change their research areas in addition to new faculty hires. The larger the current dataset (D) and the more likely it is to evolve (e) the more user contributions are needed to maintain the evolving dataset. Therefore, we could intentionally limit the size of the current dataset along with the ground truth to match the number of expected user contributions.

The expected number of user contributions over a period of time (N) is defined by the number of visitors (V) multiplied by the conversion rate (r) multiplied by the average number of contributions per visitor (E), see Equation 1.17. We can use equation 1.17 to describe the number of accurate data points contributed as $(V \cdot r \cdot E) \cdot \alpha$, and the number of inaccurate data points contributed (i.e., mistakes) is $(V \cdot r \cdot E) \cdot (1 - \alpha)$.

$$N = V \cdot r \cdot E \tag{1.17}$$

To maintain an evolving dataset, we need enough visitors to contribute enough accurate data ($N \cdot \alpha$) to overcome visitors' mistakes ($N \cdot (1 - \alpha)$), outdated existing data ($(e \cdot D)$), and new data that needs to be added (U). Equation 1.18 describes this constraint, the number of accurate contributions needed to maintain an evolving dataset. In other words, you hope the portion of visitors (V) who will make enough contributions on average to the current dataset (D) to overcome the number of data points that evolve.

$$N \cdot \alpha > N \cdot (1 - \alpha) + (e \cdot D) + U \tag{1.18}$$

We can start with Equation 1.18 to determine the lower bound for the accuracy needed to maintain an evolving dataset.

$$\alpha > 0.5 + 0.5 \left(\frac{(e \cdot D) + U}{N} \right) \tag{1.19}$$

If the dataset is static (i.e., it has no evolving data), then $(e \cdot D) + U = 0$. In this scenario, the lower bound of accuracy for a static dataset using Equation 1.19 is $\alpha > 0.5$. This aligns with our previous observations from Section 1.2.4, where if the accuracy of contributions to an existing static dataset is less than 50%, then the accuracy of that dataset would eventually approach 0%. What makes an evolving dataset different and challenging to evaluate is the inclusion of existing data that becomes outdated ($(e \cdot D)$) and new data that needs to be added (U).

Building Equation 1.19, we can find the lower bound for the number of contributions (N) needed to sustain an evolving dataset. Assuming the accuracy of contributions has to be between 0 and 1 ($0 \geq \alpha \leq 1$), we arrive at Equation 1.20.

$$N > (e \cdot D) + U \tag{1.20}$$

If Equation 1.21 is violated, then that also violates Equation 1.18, which describes the number of accurate contributions needed to maintain an evolving dataset. We can substitute $(V \cdot r \cdot E)$ for the number of contributions (N) to create Equation 1.21. We can then use Equation 1.21 to identify the lower bounds of the number of visitors (V), conversion rate per visitor (r), or the average number of edits (E) required to maintain an evolving dataset, see

Equations 1.22, 1.23, and 1.24 respectively. These lower bounds will help you know if your public data system is attracting enough visitors and if you are motivating enough visitors to make enough contributions to maintain your evolving dataset. Improving just one V, r, or E will improve the correctness of your evolving dataset. For example, maybe you conducted a pilot study where you have an estimate of the conversion rate per visitor (r) and the average number of edits (E). You can input these observations into Equation 1.22 along with estimates for your outdated existing data ($(e \cdot D)$) and new data that needs to be added (U) to predict the number of visitors you need to attract. Then you can focus on advertising and attracting enough visitors. Or maybe you have a very high number of visitors, who when they do contribute, make a high number of contributions. In this scenario, which resembles Wikipedia's ecosystem, you can use Equation 1.23 to understand if you are converting enough visitors into contributors.

$$(V \cdot r \cdot E) > (e \cdot D) + U \tag{1.21}$$

$$V > \frac{(e \cdot D) + U}{r \cdot E} \tag{1.22}$$

$$r > \frac{(e \cdot D) + U}{V \cdot E} \tag{1.23}$$

$$E > \frac{(e \cdot D) + U}{V \cdot r} \tag{1.24}$$

Equation 1.21 can be used to understand how many evolving data points Drafty's visitors could have maintained in a specific year. We can use Drafty's visitor data from the calendar year 2022 to estimate the impact 2022's visitors could have made on maintaining Drafty's evolving data. Drafty attracted more visitors and contributions in 2022 than any other calendar year; see Chapter 7. Because of the lack of new rows added in previous years, as outlined in Chapter 5, there was a large amount of evolved data from prior years that needed to be added. Thus, there were many opportunities for people to contribute compared to the scenario where we started 2022 with a 100% correct tabular dataset.

Drafty's visitors could maintain 6,191 data points in 2022. There were 14,245 total visitors in 2022 who made more than one interaction (V), the conversion rate (r) was 4.0%, and the visitors who contributed made an average of 11 edits per visitor (E). These 2022 contributions could account for a maximum of over 1,000 possible new rows or around 1 data point per existing row. This shows how increasing user motivation (i.e., conversion rate) to elicit accurate contributions can help maintain an evolving dataset such as Drafty's Computer Science professor profiles.

Overall, this shows at its maximum reported number of contributions, Drafty's community can overcome how its data evolves. However, there is still future research to conduct. For example, we should ideally control how data evolves over specified periods by beginning the time period with a 100% accurate dataset. Then over time, study how long it takes for a new data point to be added or corrected. This would require scraping the internet to determine during what time period a data point needed to be added or changed. Even current large language models do not have access to recent data. More importantly, the better you can estimate how your data evolves (i.e., the number of data points that have evolved and those that require user contributions to correct), the better you can understand if you are attracting enough visitors and motivating enough accurate contributions per visit to maintain an evolving dataset.

1.4 Overview of Contributions

This thesis covers prior research on building and deploying public data systems and developing methods to motivate more accurate crowd contributions to maintain evolving information over seven years. This research spans short and longitudinal studies focusing on how paid contributions from crowdworkers and unpaid contributions from small peer production communities affect data quality. The public data systems are freely accessible, and the code is open-sourced. Each system has functionality that allows the public to view others' contributions. Most of this thesis has been previously published in journals and conference papers. Chapter 2 provides an overview of related work focusing on crowdsourcing, peer production, and how users are motivated to contribute within related real-world systems and studies. The majority of chapter 2 contains writing that is summarized and reworked from previously published papers that I was the first author on [223, 224, 225]. The following are summaries of contributions and attributions to my co-authors who collaborated on research featured in other chapters.

Drafty: Enlisting Users to be Editors who Maintain Structured Data

Shaun Wallace*, Lucy Van Kleunen, Marianne Aubin-Le Quere, Abraham Peterkin, Yirui Huang, and Jeff Huang. In Proceedings of the Fifth AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2017).*

This research featured in chapter 3 develops and validates a model (the User Interest Profile) to automatically predict user interest in rows of tabular data based on an individual's implicit interactions with a spreadsheet interface. It also develops and integrates the User Interest Profile into a public data system, Drafty, a spreadsheet web application that stores the academic profiles of Computer Science faculty. Drafty uses the User Interest Profile to automatically ask unpaid contributors to review and correct data relevant to their interests, thus increasing an evolving dataset's overall accuracy. Drafty accomplishes this without having an unpaid contributor disclose information or perform specific tasks. In a 7-month study in the wild, this research shows that unpaid contributors asked to fix data matching their interests are three times more likely to submit accurate data than unpaid contributors asked to fix data they are uninterested in.

Crowd-based Verification Strategies for Accurate Tabular Data

Shaun Wallace*, Alexandra Papoutsaki, Hua Guo, and Jeff Huang. (Not currently published)*

While Drafty shows initial evidence of the effectiveness of unpaid contributors maintaining existing data, this line of research has yet to study the effectiveness of paid crowdworkers contributing to existing data. This research featured in chapter 4 conducts a naturalistic study of students acting as requesters in a university-level Human-Computer Interaction seminar to employ paid crowdworkers to collect and verify a tabular dataset of Computer Science faculty profiles. It serves as the first attempt in this dissertation to study the accuracy and behaviors of paid crowdworkers editing tabular data. While this research was never published, the authors who contributed to it should receive attribution within this dissertation.

By observing quantitative and qualitative evidence from novice requesters and paid crowdworkers, this research develops recommendations for new tabular data verification strategies to employ paid crowdworkers to improve data quality. The findings from this research are the precursor to chapter 5's case study on paid crowdworkers. This research shows initial evidence that recruiting trusted crowdworkers for additional tasks is more accurate at fixing data requiring domain-specific knowledge. For example, this research shows that the accuracy of strategies differs across data types requiring varying levels of effort and domain-specific knowledge. The research also shows that using crowdsourcing platform's preset filters to recruit workers, like Amazon Mechanical Turk's Masters Qualifications, did not significantly increase, and in some cases decreased, the accuracy of contributions. The naturalistic observations and findings from this research contribute to a more controlled case study to evaluate paid crowdworkers in chapter 5.

Case Studies on the Motivation and Performance of Contributors Who Verify and Maintain In-Flux Tabular Datasets

***Shaun Wallace**, Alexandra Papoutsaki, Neilly H. Tan, Hua Guo, Jeff Huang. Proceedings of the ACM on Human-Computer Interaction 5, (CSCW 2021).*

While running a 7-month study in Drafty showed that users interested in the data could make accurate contributions, it does not study the editing behaviors of paid crowdworkers. Also, since the original Drafty study was only 7 months, it did not study unpaid contributors' effectiveness and editing behaviors over multiple years. Thus, it does not cover temporal events such as new hires and retirements that will decay the data. Also, this longitudinal work does not actively advertise Drafty. Thus, what type of visitors, contributors, and behaviors does it attract? The research answers many of these questions by conducting two case studies: one studying the accuracy of contributions from paid crowdworkers over 2 short weeks and another studying the accuracy of contributions from unpaid contributors over 2.5 years. Chapter 5 describes these case studies. This publication refers to a pilot study to develop paid verification strategies. Chapter 4 of this dissertation is the full study featuring quantitative and qualitative analysis used to develop these paid verification strategies.

This study builds on the work from chapter 4 to develop five new tabular data verification strategies for data curators to employ paid crowdworkers to improve data quality. Across the five paid verification strategies, *Expert Rule* controls costs the best. It generated accurate contributions at less than half the cost-per-contribution compared to the second most accurate strategy *Find-Fix-Verify*. Also, *Expert Rule* uses a low-cost redundant data collection step first, followed by a final verification from a trusted crowdworker. The paid verification strategies recruiting a trusted crowdworker produced more accurate contributions for data types requiring domain-specific knowledge. The strategies employing paid crowdworkers produced contributions that were 64% accurate. In contrast, the strategies that waited on unpaid contributors produced contributions that were 89% accurate. The two case studies also show that unpaid contributors interested in the data make accurate contributions per data type (column), regardless of the domain-specific knowledge necessary to interpret subjective data. In contrast, paid crowdworkers made more inaccurate contributions for data requiring domain-specific knowledge. Also, unpaid users interested in the public data made accurate contributions without signs of vandalism, regardless of the number of visits or interactions with a public data system. In summary, unpaid contributors could make accurate contributions regardless of their prior experience with a public data system. These results lead to the idea that to maintain evolving information in a public data system, attracting a large number of interested users is better than recruiting a small number of users to make the majority of contributions.

Sketchy: Drawing Inspiration from the Crowd

Shaun Wallace*, Brendan Le, Luis A Leiva, Aman Haq, Ari Kintisch, Gabrielle Bufrem, Linda Chang, and Jeff Huang. Proceedings of the ACM on Human-Computer Interaction 4, (CSCW 2020).*

While the prior research on Drafty and paid crowdworkers focus on data that evolves annually, this research studies data that evolves rapidly over minutes. This previously published research featured in chapter 6 develops the public data system Sketchy, where students join virtual rooms to sketch, ideate, and freely peek at their peers' in-progress sketches using the Peek feature. When unpaid contributors use Sketchy's digital Peek functionality during real-world sketching tasks, we found that the Peek feature is key to increasing users' contributions in the form of more detailed sketches, overall creativity, and satisfaction with their final sketch. Also, the results from this research show that providing unpaid users with the agency to seek and engage with sketching examples created by other users motivates them to contribute information back to the public data system over short periods. This research also shows initial evidence that when users prompt the system to recommend an example from the data (i.e., the Peek feature), they prefer recommendations that are selected randomly compared to recommendations that are similar to the most previous example.

Life Support for Maintaining an Evolving Tabular Dataset: Attract-Engage-Motivate Crowd Contributions

Shaun Wallace*, Diana Na Kyoung Lee, Zhengyi Peng, Long Do, Talie Massachi, Alice Marbach, Jiaqi Su, David B Miller, and Jeff Huang. (Will be Submitted Soon).*

The last two chapters of this dissertation (Chapter 7 and Chapter 8) will be submitted for review and possible publication. Chapter 7 contributes a new version of Drafty that includes the Javascript library PolyMu to handle infinite scrolling and allow large tabular datasets stored as HTML to be search indexed. In this chapter, we also develop two public systems that use Drafty's data: CS Open Rankings and the Databaits API. Both systems dynamically generate insights from Drafty evolving, making these publicly available to source (i.e., attract) visitors back to Drafty. Visitors to Drafty from these "dynamic" sources make more contributions at a higher rate of accuracy than "organic" visitors arriving from unknown sources such as search engines. Notably, visitors to Drafty from these "dynamic" sources also make more contributions with the same accuracy as visitors from the computer science community who we "asked" to contribute to Drafty. CS Open Rankings and the Databaits API provide additions to Drafty that continuously attract high-quality visitors but take little to no effort to maintain compared to asking people on Twitter or posting on forums. The Databaits API is also integrated into Drafty's spreadsheet interface as the "did you know" feature where any user can freely engage with automatically generated insights about Drafty's Computer Science professor dataset. Users who see an insight in "did you know" are six times more likely to generate a random insight than an insight similar to one they previously saw. Users who engage insights generated by the Databaits API in the "did you know" feature are more likely to stay engaged and contribute to Drafty than those who do not. Chapter 8 features a survey-based study where paid crowdworkers and everyday visitors (i.e., unpaid contributors) to Drafty freely choose to complete a discrete choice experiment. Compared to the naturalistic study conducted in Chapter 7, this discrete choice experiment directory compares the accuracy and motivations of paid crowdworkers and unpaid contributors. When editing data, everyday unpaid contributors visiting Drafty compared to paid crowdworkers were 1.9 times more accurate for difficult to find data and 1.5 times more accurate for data requiring domain-specific expertise. While paid crowdworkers and unpaid contributors are mainly motivated by pay level, estimated time to complete, and their perceptions of tasks (is it interesting and will it help people) to contribute, they make different trade-offs when selecting between tasks. Highly accurate unpaid contributors prefer tasks where they collaborate with others, while paid crowdworkers prefer tasks where they can collaborate with Artificial Intelligence. Paid crowdworkers are more willing to complete tasks if the tasks pay more and are quick to complete. Whereas unpaid contributors are willing to do tasks for less or no monetary compensation if they match interests, their contribution helps others, or they contribute their specialized knowledge. Overall, the universal factors motivating contributors and paid crowdworkers mirror the design of this dissertation's public data systems.

Chapter 2

Related Work

Dataset curators, also called requesters, often recruit paid crowdworkers by posting short, repeatable micro-tasks to quickly collect information for beneficial datasets [65, 123]. Historically decomposing complex tasks into simpler ones aligns with the idea of piecework [5]. This chapter presents an overview of recent research related to crowd-powered systems, paid crowdsourcing, peer production, learnersourcing, and methods to motivate and improve the quality of crowd contributions that are simple in execution but complex in nature.

2.1 Crowd-Powered Systems

Jeff Howe coined the term crowdsourcing [102] as "an umbrella term for a highly varied group of approaches that share one obvious attribute in common: they all depend on some contribution from the crowd." [103]. He also categorized four main applications of crowdsourcing: 1) crowd wisdom or collective intelligence, 2) crowd creation or user-generated content, 3) crowd voting, and 4) crowdfunding. This dissertation combines the first two categories, allowing users to freely use their collective intelligence to contribute to public data systems.

Traditional crowdsourcing platforms, such as Amazon Mechanical Turk, support an ecosystem where requesters can post micro-tasks for many use cases, such as data collection and verification. While this is practical to generate an initial dataset [173], the long-term upkeep of the dataset can prove exponentially challenging. Crowdfill [175] is a crowdsourced system that maintains structured datasets using the microtask-based approach of Amazon Mechanical Turk. Rather than giving each worker a set of tasks to complete, it presents paid crowdworkers with one shared data table they can fill in however they want. Workers can also rate data submitted by others. This approach plays to the individual strengths of paid crowdworkers and results in higher-quality submissions. In Crowdfill, Park and Widom mention that the system could potentially be improved by automatically recommending specific pieces of data to individuals based on their skills. This dissertation's initial versions of the Drafty system explore this idea, matching unpaid users to fix data that match their interests. Crowdfill's model and other similar systems, such as Wisteria [87], may be hard to sustain long term. Repeatable tasks for maintaining data through Amazon Mechanical Turk are impractical due to increased cost, time, and accuracy risks [152]. These tasks and costs to employ crowd workers increase as the dataset grows. By developing systems that rely on unpaid users, it is possible to overcome these constraints of traditionally paid crowdsourcing.

There is a history of crowd-powered systems that seek to solve long-standing issues in crowdsourcing, such as bringing the benefits of crowd-powered work to an inexperienced audience. Soylent [22] and Fantasktic [85] are novice-centric systems built to address common mistakes when crowdsourcing. For example, providing insufficient guidance to workers or not verifying the data. This dissertation explores systems that allow it addresses both of these shortcomings and does not require a requester to post micro-tasks for workers. Drafty empowers its users to perform tasks in an undirected manner, similar to the find and fix or find and verify pattern in Soylent. Previous research also shows little difference between expert and non-expert workers for routine tasks, but this changes for specialized tasks [202]. Research on knowledge-intensive tasks [61] further supports the idea that crowds with specific knowledge [86, 169] provide more accurate contributions to data requiring domain-specific knowledge. This dissertation seeks to solve this problem by exploring if users interested in a specific dataset or task are more accurate at editing data for specialized fields. Crowdfill, Soylent, and many other crowd-powered systems have not assessed sustainable, low-cost solutions to this problem. While many systems use paid crowdsourcing approaches to motivate and create contributions, there is a long history of people contributing within peer production systems.

2.2 Peer Production

Peer production systems often feature decentralized task designs, communities of users, community ownership of contributions, and freedom (i.e., user agency) to empower its users to make crowd contributions [16]. For example, research in peer production systems explores utilizing user interactions and interests for unstructured data upkeep. SuggestBot uses Wikipedia editors' contribution histories to suggest editing tasks [53]. WikiTasks supports the creation of site-wide tasks and self-selection of personal tasks within Wikipedia [127]. Unlike SuggestBot and WikiTasks, Drafty hosts a structured dataset of tabular data rather than unstructured Wikipedia articles. This distinction allows data to have pre-specified attributes and to track a different set of interactions. Also, Drafty and Sketchy are custom-built systems that are not dependent on another platform, such as Wikipedia, to work effectively.

This dissertation explores automated mechanisms to infer interests from interactions to make recommendations from the data and how users respond to these recommendations. For example, Drafty automatically selects the rows and individual data points for users to fix. In contrast, SuggestBot relies on downloading and analyzing large sets of Wikipedia articles. Additionally, WikiTasks relies on humans to manually create tasks [53]. Drafty records interactions beyond those used by SuggestBot (search, click, sort), which allows Drafty to build interest profiles and study user engagement using a wider variety of interactions. This allows for more robust models and analysis to enhance system recommendations and study implicit human behaviors during unpaid contributions. SuggestBot, Kylin, and other work from MovieLens evaluate whether they increase submission rates of edits [8, 53, 242]. This dissertation evaluates this and the interactions that increase the rates and accuracy of contributions. A research area focusing on crowd contribution research that combines aspects of crowdsourcing and peer production is learnersourcing.

2.3 Learnersourcing

Research in learnersourcing has explored using interactions from native system users to assist with the upkeep of online content. Williams et al. developed AXIS, a system that combines feedback from learners in online courses with machine learning algorithms to improve problem explanations [232]. Future students in the course use these explanations for assistance and engage in the exact feedback mechanism. This loop over time helps these explanations adapt to new user habits through an automated system. The long-term benefits of this feedback mechanism constitute a significant motivation for Drafty's integration of user interest profiles to solicit unpaid users to maintain data. To perform such complex analysis, Drafty relies on collecting and inferring interest from the interactions of its users. In a similar endeavor, Kim et al. collect interaction data from online learners and use it to improve the experience of navigating video lectures [121]. Li et al. investigate how to optimally identify groups of workers based on their characteristics [137]. Their method identifies the subgroup of workers best suited for a particular task for each task. This dissertation explores targeting individuals rather than groups of users. Also, this dissertation explores groups of anonymous users in Drafty and Sketchy. Thus by treating users as individuals, the systems can hopefully provide a better-individuated experience without requiring personally identifiable information. This approach aligns with recent ethical concerns in learnersourcing, where people should not be targeted based on demographics and should have more ownership over their contributions [208]. Drafty and Sketchy are developed over time to provide users agency over their interactions and provide randomized examples not based on potentially biased algorithms per Singh et al.'s recommendations.

2.4 Recommendation Systems

This dissertation explores integrating features that solicit or provide the user agency to seek timely examples. This interaction mirrors research in recommendation systems. For example, the initial versions of Drafty use the user interest profile to target users for data upkeep. Targeting users based upon surveys, interactions, preset tasks, or interests is a common theme among recommender systems [190]. These interactive and intelligent systems provide recommendations that match users' preferences [184]. Drafty builds on this idea to assess if soliciting users using their interests motivates them to volunteer to review more data and do it more accurately. While Sketchy and later versions of Drafty explore providing users agency when they seek and interact with a system-generated recommendation.

GroupLens [188] is a collaborative filtering system that predicts a reader's preference for an article based on ratings. This is accomplished through explicit feedback provided by the user, but implicit feedback from user interactions can be similarly valuable [45, 104]. Huang et al. collected fine-grained interaction data in the wild to understand web users' behaviors and search patterns. Chen et al. also explore behavioral targeting in web users primarily using clicks. These behavioral interactions have shown positive results in building user interest profiles in related recommender systems [243]. Zhao et al. show how new content or data created by users can be used to enhance user interest profiles in recommender systems. The initial version of Drafty builds on those ideas by inferring interest from users automatically from easy-to-collect online interactions. This dissertation also explores how small communities of anonymous users sharing similar interests often seek and respond to recommendations differently than traditional research in collaborative filtering and recommendation systems.

2.5 Approaches to Source and Improve Crowd Contributions

2.5.1 Paid Approaches

Paying crowdworkers to verify crowdsourced datasets is a standard method to improve their accuracy [180]. For example, crowdworkers validated alt-text tags to improve the usability of social media posts for blind users [197] and collected a dataset to help researchers understand personality disorders [155]. This dissertation builds on these ideas of paying crowdworkers to improve data quality by combining novice requester behaviors and classic crowdsourcing approaches to generate five verification strategies for correcting tabular datasets in chapters 4 and 5. Chapter 5 adapts Bernstein et al.'s popular Find-Fix-Verify method [21] and Hirth et al.'s Majority Decision [97]. Chapter 5 also introduce variations of them inspired by Marcus' tactics for knowledge-specific tasks [151]. Balancing the speed and cost of recruitment in these strategies is difficult. Huang and Bigham developed the Ignition Framework combining on-demand recruiting and the retainer model to balance recruiting workers' cost and immediacy [106]. While it proved successful, this method would require large sums of money to pay crowdworkers to improve evolving information within a public data system over time. Another approach is to recruit paid crowdworkers in a retainer pool for quick access [20].

Past work focused on methods to recruit paid crowdworkers to solve specialized tasks. Kittur et al. [124] increased the quality of solving complex problems by using simpler micro-tasks. Likewise, in Turkomatic, crowdworkers simplify complicated steps recursively; then, other crowdworkers verify solutions and form an answer [129]. These approaches require extra time for requesters to simplify tasks and manage multiple crowdworkers. Previous efforts also tried to match crowdworkers to micro-tasks matching their skills and expertise [159]. This dissertation work extends these ideas by allowing requesters to recruit specific crowdworkers to perform verification tasks.

2.5.2 Unpaid Approaches

A second approach to improve the quality of crowdsourced datasets is to rely on unpaid contributors' interests and expertise to ensure accurate and continuous improvements. Xu and Maitland [236] employed similar concepts while studying users' participatory data maintenance in field trials with urban refugees. Quattrone et al. [179] studied geographic maintenance practices within OpenStreetMap, where users freely update spatial information. Chapter 3 relies on user interest in information to attract and solicit contributions within a tabular dataset. While this short-term study also identified domain-specific challenges in data maintenance, it did not capture edits where datasets change naturally over time. This dissertation builds on these past ideas by studying visitors' interactions with a tabular dataset while acting as unpaid contributors who edit data over time.

Past work from recommender systems shows that building user interest profiles using implicit feedback from user interactions can successfully target users [243]. Other work focuses on unpaid approaches to target users. For example, peer production systems, such as SuggestBot [53] and Wikitasks [127] utilize user interactions and edit histories to suggest edits and assist in task design. In contrast to Wikipedia [181], where most content is long-form articles and some small tables, this dissertation focuses on users editing evolving tabular data consisting of thousands of rows in spreadsheets. Also, in chapter 6, users make individual contributions in the form of sketches that are automatically available for other users to view.

2.5.3 Improving Domain-Specific Knowledge

A benefit of a public data system is potentially attracting a group of like-minded users. This anonymous community could possess greater domain-specific knowledge than paid crowdworkers within the same system. For example, maintaining data over long periods poses unique challenges for soliciting edits, especially for domain-specific data. A simple approach is to continuously post paid micro-tasks for knowledge-intensive tasks [86, 169] or use custom systems [87]. For example, in Crowdfill [175], crowdworkers freely edit a spreadsheet and up/down-vote possible data points. This closed system gives crowdworkers freedom but does not allow requesters to employ verification tasks freely. While paying crowdworkers is an effective solution for short-term verification, it can become less practical over time due to financial constraints, time, and data integrity [152].

2.5.4 Motivating Paid and Unpaid Contributors

Self-determination theory defines two types of motivation: intrinsic and extrinsic [63]. Intrinsic motivation refers to an individual's inherent desire to participate in an activity, whereas conditional rewards influence extrinsic motivation (e.g., money) [13, 194]. This dissertation does not seek to directly evaluate every possible motivation that could influence paid crowdworkers and unpaid contributors. However, it is essential to acknowledge related work and to understand how diverse and similar user motivations can be among crowd contributors.

Historically, payment per task can extrinsically motivate crowdworkers [98]. Also, meta-incentives can augment extrinsic motivation for future payments. For example, rejecting low-quality work can negatively affect a paid crowdworker's approval rate, thus impacting their ability to complete future work [150]. In chapters 4 and 5, requesters had to balance these constraints and determine appropriate compensation schemes per task to ensure quality work. While paying crowdworkers is useful for collecting data, there are concerns over accuracy. Reasons for this can include varying motivation [162] and effort levels between crowdworkers [21], malicious crowdworkers [76], and insufficient expertise [202].

People may be intrinsically motivated to contribute their time toward causes they are interested in [159, 160]. For example, chapters 3 and 5 feature strategies that rely on visitors to a tabular dataset making unpaid contributions. We posit they might participate because of an interest in Computer Science. This reliance on user interest has proven successful in maintaining a popular resource in the CS community, CSRankings [17]. To the best of my knowledge, there is no equivalent system relying on paid crowdworkers for contributions.

Previous research has explored hybrid models simultaneously providing users with intrinsic and extrinsic motivators to contribute. In prior work by Flores-Saviaga et al., unpaid contributors were more effective at open-ended tasks, such as original content creation. At the same time, paid crowdworkers were more effective at completing more straightforward tasks following strict guidelines [31, 73]. As a result, they suggest a hybrid approach: leveraging volunteer work for original content and employing crowdsourced work to structure and prepare content for real-world use. This dissertation provides new insights about potential hybrid models by studying the results of two separate case studies in chapter 5 where paid and unpaid users complete similar tasks. In addition, chapter 6 focuses on people making unpaid contributions to rapidly create and update original content in the form of sketches.

2.6 Conclusion

This dissertation explores developing public data systems with features for a specific community of users with similar interests in its data. These like-minded anonymous communities pose several potential benefits, notably cost-effective continuous engagement with the system. By building systems using a hybrid approach between crowdsourcing and peer production, this dissertation builds on existing research by exploring new methods to improve the quality of crowd contributions from paid crowdworkers and unpaid contributors. The trade-offs crowd contributors make when choosing which task complete is an underdeveloped research area. This dissertation explores these trade-offs and grounds them in a theoretical framework for maintaining evolving information using public data systems and integrative hybrid approaches.

Chapter 3

Drafty: Enlisting Users to be Contributors who Maintain Tabular Data

This chapter presents Drafty, a public data system that uses a spreadsheet interface to allow visitors to edit the academic profiles of tenure-track Computer Science faculty freely. Drafty tracks user interactions with the spreadsheet to create a user interest profile per user. Drafty asks random users to review and possibly fix data matching their interests. This chapter is substantially similar to [225], where I was the first author and responsible for all aspects of the user studies, system design and implementation, analysis, and most of the writing. As part of the contributions of this chapter, Drafty is open-sourced as a public data system[1].

3.1 Introduction

Much of the data around us constantly changes and needs updates to stay accurate and relevant. Popular examples of prior datasets include company management information Crunchbase [56]), legal filings, soccer player career history (e.g., Crowdfill [175]), sponsored funding opportunities, socio-economic and law enforcement data (e.g., Communities and Crime Dataset [185]), and country-level health statistics. Data may change for many reasons: new events, personnel changes, or existing data is revised. Without proper upkeep, the accuracy of data can decay over time. This data often requires more monitoring and upkeep than one individual can handle.

A common approach is relying on traditional paid crowdsourcing efforts using platforms such as Amazon Mechanical Turk, which are insufficient in several ways. First, paid crowdworkers may not be familiar with the specialized vocabulary used in start-ups or academia. They might not have the necessary context to provide accurate informa-

[1] Drafty is available at http://drafty.cs.brown.edu/

tion [173], as revisions to this dataset require domain-specific knowledge. Second, systems powered by Amazon Mechanical Turk to upkeep tabular data pose long-term financial commitments that might not be sustainable.

This chapter develops the first version of the public data system Drafty. Drafty builds upon the fundamental ideas of Crowdfill, but it does not rely on paid crowdworkers, like in peer production. Instead relies on *drafting* the regular users who visit the data to become contributors. Drafty empowers the user by enabling them to edit data freely. It determines correct data using a Wikipedia-like. Hence, these users who freely visit Drafty are called "unpaid contributors." Most importantly, Drafty is initially designed to harness the interests of its unpaid contributors to maintain a tabular dataset. It builds user interest profiles to target data points for unpaid contributors to review and correct that match their interests. Prior research supports the idea that crowdworkers are more willing to do higher quality work with better retention if the task is relevant to their interests [51].

A user interest profile is automatically constructed for each unpaid contributor based on their interactions with Drafty's interface. These interactions are collected remotely without disturbing the user (e.g., using methods from [104]). We hypothesize that an interaction-based profiling approach provides higher-quality data and reduces long-term costs for maintaining tabular datasets. As part of the system design of Drafty, the research challenge focuses on how to construct user interest profiles to use them to match unpaid contributors to data they could best review.

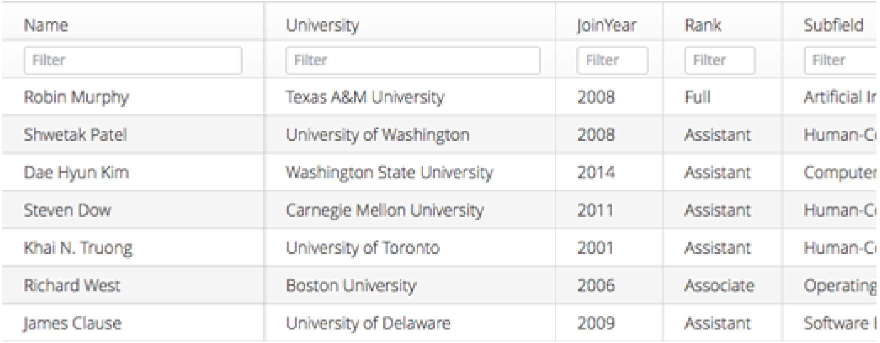
The research contributions of this chapter are twofold: (1) we describe the implementation of Drafty, a public data system that hosts tabular datasets that are self-sustaining, capturing and interpreting user edits, clicks, searches, and column sorting, and (2) we validate the ability of automatically-generated user profiles to reflect user interest and show that users who are asked to review data that match their interests are more likely to volunteer and also provide more accurate suggestions.

While Drafty is a data-agnostic platform for demonstration and experimentation, it was populated with a dataset consisting of over 50,000 data entries that are used to build academic profiles of over 3,600 computer science professors from across the USA and Canada[2]. This data came from Mechanical Turk crowdworkers recruited to build the dataset as part of an assignment in a Human-Computer Interaction seminar. The original data collection was presented in research from Papoutsaki et al. [173].

Each professor's academic profile included: their affiliated university, the year they joined as faculty, their rank, subfield area of expertise, where they received their Bachelors, Masters, and Ph.D. degrees, where they did their PostDoc, a link to a profile photo, and links to sources for the aforementioned information types.

This dataset received reviews and corrections from over 50,000 visitors from 2014 to 2017. Hence, the current quality and maturity of the dataset are likely representative of other tabular datasets.

[2] This data is available at http://drafty.cs.brown.edu/professors/



Name	University	JoinYear	Rank	Subfield
Filter	Filter	Filter	Filter	Filter
Robin Murphy	Texas A&M University	2008	Full	Artificial Ir
Shwetak Patel	University of Washington	2008	Assistant	Human-C
Dae Hyun Kim	Washington State University	2014	Assistant	Computer
Steven Dow	Carnegie Mellon University	2011	Assistant	Human-C
Khai N. Truong	University of Toronto	2001	Assistant	Human-C
Richard West	Boston University	2006	Associate	Operating
James Clause	University of Delaware	2009	Assistant	Software I

Figure 3.1: Screenshot of the original Drafty spreadsheet interface (from 2016) populated with profiles of computer science professors.

3.2 Drafty Public Data System

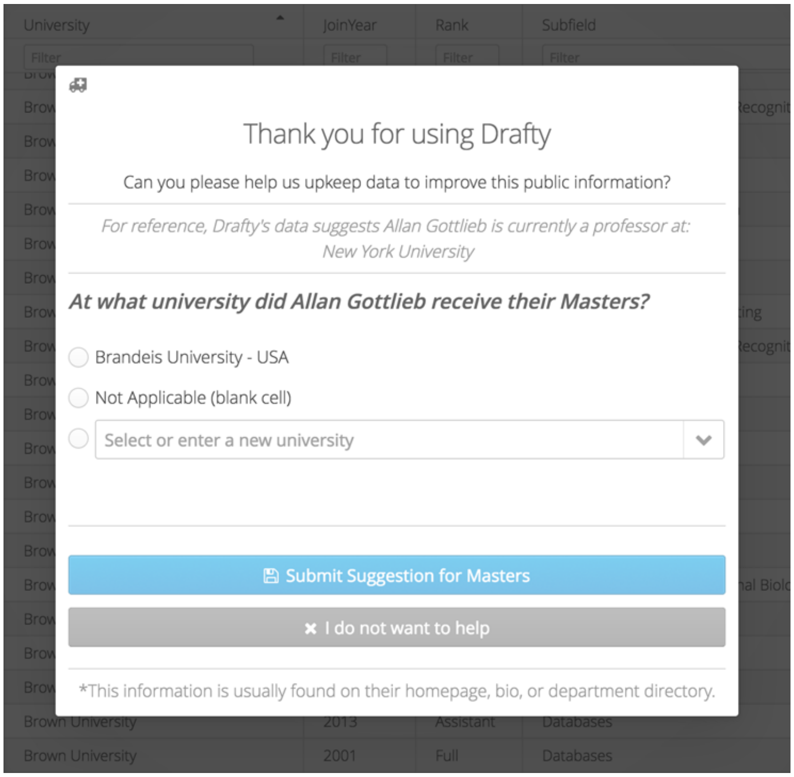
Drafty is an interactive online public data system for building and maintaining large tabular datasets. It leverages the expertise of its users by drafting them to become contributors. Drafty's user interface and basic functionality closely resemble that of a spreadsheet application (Figure 3.1), providing unpaid contributors a minimal learning curve. This is in contrast to traditional database systems (e.g., Freebase [24] and Wikidata [220]) that present structured data as articles and require switching to an editor mode. Unpaid contributors can freely submit new data or evaluate Drafty's existing data. Drafty uses interactions to provide a human-in-the-loop mechanism to allow normal users to edit data more effectively. Previous studies have demonstrated that data quality increases if users are aware that others will review their work [105]. Drafty solicits unpaid contributors to help validate or fix data using an additive model for interaction-based profiling (Figure 3.2). This model is the user interest profile.

The more interactions an unpaid contributor performs their user interest profile becomes more robust, and Drafty can better personalize requests to review data. Other research has validated methods for inferring interest from user interactions to build interest profiles [118]. The benefit of this system is that over time, solicited and unsolicited data fixes lead to a more accurate and mature dataset. If unpaid contributors trust the data found in Drafty and feel empowered to correct inconsistencies, their long-term commitment to the system should increase.

The following sections describe Drafty's methods for assigning confidence scores per data suggestion, user interest profiling, determining data for review, and a user study to validate the user interest profile.

3.2.1 Confidence Score per Cell

This version of Drafty uses a simple method to assign confidence scores per cell. The most recent submitted value per cell is assigned the highest confidence score among all possible values for that cell. The reasoning for this decision is twofold. First, at this study's onset, Drafty did not have enough data to create a valid learned model for assigning confidence scores. Therefore, the initial decision is to treat unpaid contributors similarly to administrators on


University
JoinYear
Rank
Subfield
Filter
Filter
Filter
Filter
Submit Suggestion for Masters
✖ I do not want to help
Brown University 2013 Assistant Databases
Brown University 2001 Full Databases

Thank you for using Drafty

Can you please help us upkeep data to improve this public information?

For reference, Drafty's data suggests Allan Gottlieb is currently a professor at: New York University

At what university did Allan Gottlieb receive their Masters?

○ Brandeis University - USA
○ Not Applicable (blank cell)
○ Select or enter a new university

*This information is usually found on their homepage, bio, or department directory.

Figure 3.2: Drafty prompts unpaid contributors to review data matching their interest profile. They can confirm a previous suggestion, submit an alternate suggestion, or close the prompt.

Wikipedia [127]. Unpaid contributors are trusted to edit and maintain the dataset since they are interested in the content. This is the same pattern followed in a normal business environment if users collaborated on Google Sheets. Second, Drafty provides unpaid contributors with a simple interface to edit cells. They select from pre-existing suggestions or submit a new one. This is in contrast to common upward and downward voting mechanisms found in other systems like Crowdfill [175]. Drafty's method mirrors mechanisms in normal business environments and has a lower learning curve.

Another method to choose the correct value per cell from crowdsourcing is Majority Decision [97]. This method assigns the highest accuracy to the most common data point and scales well over large datasets. However, since tabular datasets change over time, it will take too many suggestions for a new data point to have the highest confidence score. Also, it could be confusing for someone to submit a new edit, and because it is uncommon, the edit will not be considered correct. This could reduce their trust and subsequent usage of the public data system.

Another viable method for selecting the most accurate data point is Majority Decision. Majority Decision [97] assigns the highest accuracy to the most common data point and scales well over large datasets. However, since tabular datasets change over time, it will take too many suggestions for a new data point to have the highest confidence score.

Category	Interaction Type	Weight
Click	Click (Highlight)	1
	Row of Click	*1
	Double Click	*1
Search	Partial Search	2
	Complete Search	3
Edit	Suggestion (Validate)	4
	Suggestion (New Data)	5

Table 3.1: Weights (T_w) per interaction type used to build user interest profiles. (*A field is awarded an additional point when clicking any field in that row.)

3.2.2 Creating a User Interest Profile per User

Each unpaid contributor has a unique profile computed for them. Interactions are recorded on the browser and stored in a central database. A user interest profile is either created or retrieved from the database when a new session starts using the unpaid contributor's browser cookie.

User Interest Weights

Drafty's user interest weights mirror other approaches to establishing implicit user interest based on browsing information. Much of the literature on building user interest profiles based on implicit feedback focuses on how users interact with the web and how to tailor web searches to pages of high interest. We used Chan's ideas about building web user profiles to inform our additive interaction model. Chan considers the frequency at which a user visits a page as the highest indicator of interest [39]. Our additive model performs similarly, where unpaid contributors are determined to be more interested in a cell they more frequently interact with. Chan also notes that if a user clicks more links on a page, the page will likely be of higher interest to that user. Similarly, Drafty reflects that if a user interacts multiple times with a category (e.g. clicks on three professors from the same university), this indicates general interest in that category.

Kim et al. outline five categories of observable behaviors users exhibit when interacting with websites [118]. In Drafty's user interest model, interactions are weighted by the interaction type, as shown in Table 3.1. Three categories adapted from their model are used to determine the weights for "Click," "Search," and "Edit." These are the main interactions that users perform on the data, increasing order of demonstrated interest. The "Click" category is the weakest form of interaction that users exhibit, as clicks may be normal user behavior and thus may not show intent. The "Search" category shows that a user exhibited an intent to engage with one or more of the results. The "Edit" category shows intent and knowledge of a precise cell, so is weighted the highest.

Recording Interactions

A **click** on a specific cell might indicate interest in that particular data point. For example, a click on a cell that contains the value “Databases” might indicate a broader interest in that subfield. However, that interaction might also indicate interest in that specific row. Other potential factors could influence user interest in that row, such as the professor’s university. Drafty handles these possibilities by first adding one point to the column’s score for the value that was clicked. Then, Drafty adds one additional point to the additional column types in that row. **Double-clicks** are given an additional point because this indicates an attempt to edit data.

A **complete search** is recorded when an unpaid contributor stops typing and leaves a search field. This indicates they are satisfied with the result. A **partial search** is a search in progress where a user pauses typing but does not leave the search field. This indicates the user is examining the results but is still in the process of searching. In both cases, the user interest profile accumulates points for all possible data values the unpaid contributor intends to search for.

Drafty records points for **edits** in two different ways. If an unpaid contributor selects an existing value of a cell, Drafty adds 4 points for the validation of the selected value. If an unpaid contributor suggests a new value for the cell, Drafty records 5 points for the relevant value. In the case of new values, Drafty considers this a stronger signal of interest given the proactive nature of the interaction.

Determining Data for Solicitation

An unpaid contributor’s interest profile is used to determine what row will be used to solicit data review (Figure 3.3). Drafty will randomly select a column from that row for data upkeep. These solicitations are triggered randomly just before 10 interactions have been made. This number was derived from the average number of interactions made before data was submitted for upkeep during the system’s pilot test phase. Drafty computes an interest score for each row in the dataset,

$$f_n(r) = \sum_{h=1}^{N} \sum_{i=0}^{T} (T_w \cdot x) \tag{3.1}$$

Where N is the total number of columns. T is the number of interaction types (i.e., click, search, etc...), T_w is the weight of an interaction type (see Table 3.1). x is the number of interactions per value. The higher the computed score, the greater the inferred interest. A score of zero indicates no interest. The scores are used to compute the probability (Equation 3.1) that the row will be selected for review by an unpaid contributor.

$$P(r) = \frac{f_n(r)}{f_o(r)} \tag{3.2}$$

$P(r)$ is the probability a row is solicited for review based on a user’s interests. $f_o(r)$ is the total user interest score for

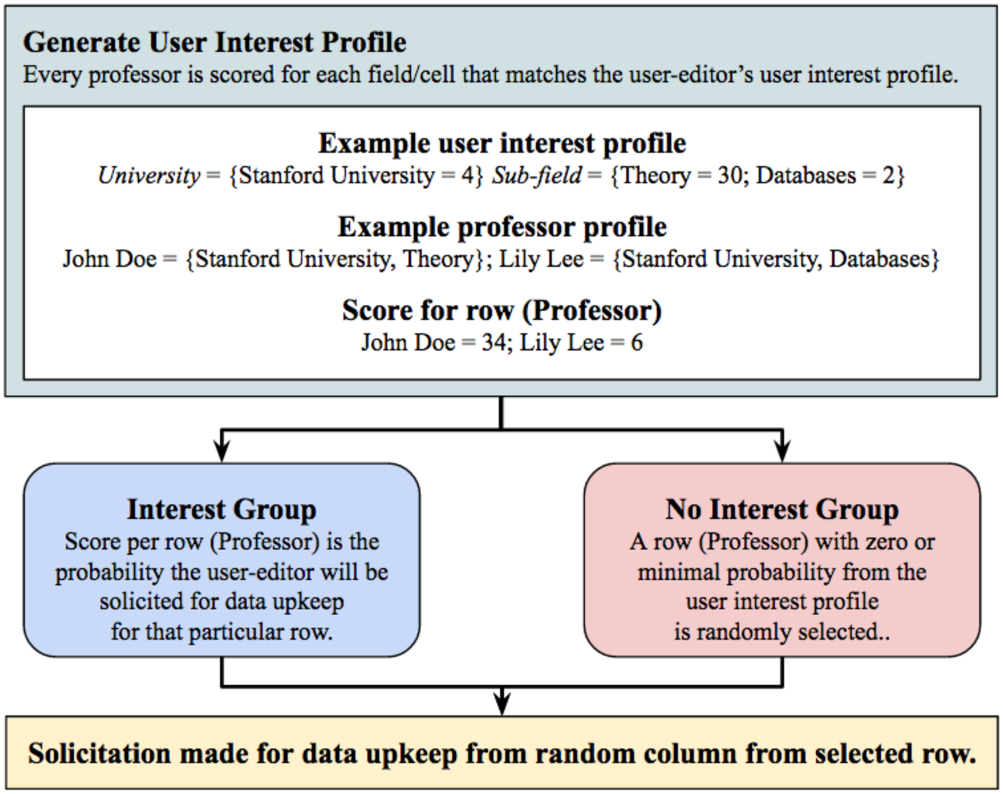

Generate User Interest Profile
Every professor is scored for each field/cell that matches the user-editor's user interest profile.
Example user interest profile
University = {Stanford University = 4} Sub-field = {Theory = 30; Databases = 2}
Example professor profile
John Doe = {Stanford University, Theory}; Lily Lee = {Stanford University, Databases}
Score for row (Professor)
John Doe = 34; Lily Lee = 6
Interest Group
Score per row (Professor) is the probability the user-editor will be solicited for data upkeep for that particular row.
No Interest Group
A row (Professor) with zero or minimal probability from the user interest profile is randomly selected..
Solicitation made for data upkeep from random column from selected row.

Figure 3.3: Drafty creates a user interest profile per user based on their implicit interaction data. Then it randomly solicits users to review and fix data based on their experimental group.

all rows. $f_n(r)$ is the user interest score per row. To solicit data for review, Drafty randomly selects a row from the user interest profile. Rows with higher interest scores have a higher probability of solicitation. This ensures a measure of randomization so the same rows are not repeatedly selected.

Extending the User Interest Profile

Drafty uses an additive model to derive user interest from interactions. This interaction-based profiling relies on a set of weights for each interaction type. These weights were iteratively developed based on qualitative observations of unpaid contributors' interactions, and feedback garnered from these sessions using the think-aloud method.

Drafty's additive profiling model calculates relative interest based on an unpaid contributor's interactions with the system. This technique relies on the assumption that over time interactions best representing unpaid contributor interest will outweigh interaction "noise", such as errant clicks on non-relevant cells. The additive model acts as a catch-all for different interaction patterns. This approach is extensible to high activity users while still providing sufficient data from low activity users. A possible extension could be weighing recent interactions more strongly to give more weight to current interests. Another possible extension is to track the peer-judged validity of an unpaid contributor's data fixes. An unpaid contributor whose suggestions are judged valid by their peers might be considered more of an expert, and their contributions are given higher confidence scores.

Another pertinent question is how to make interaction-based profiling data agnostic. For example, Drafty records interest in the university, Bachelors, and Masters separately. However, it captures interest in universities from a broader perspective. The assessment of how specific columns in a tabular dataset is dependent on other columns was not assessed in this current version of Drafty. For example, an unpaid contributor interested in "University X" in the university column might reasonably be assumed to have interest in "University X" in the Bachelors column. However, Drafty does not currently aggregate interest in the model because that relationship might be too specific to this dataset.

Drafty's current iteration faced a few technical limitations. While the initial click on a cell is recorded, actions such as copy and paste cannot be extracted currently due to framework limitations. Drafty is also unable to save scrolling actions. Therefore, while Drafty is aware of the set of rows currently visible, it is not apparent which rows were actively visible on the screen.

3.2.3 Validating the User Interest Profile

A validation study was conducted to assess the validity of the user interest profile. The procedures were approved by our Human Subjects Protection (IRB) office. Following established human subject guidelines, all participants consented to the study. Participants were specifically recruited via convenience sampling from the computer science community, comprising 20 undergraduate students, graduate students, faculty, and staff with academic and professional backgrounds in Computer Science. Participants had current and past experiences with 20 different universities. Their range of expertise in Computer Science consisted of 16 different subfields. Participants included 8 (40%) females and 12 (60%) males with a mean age of 29 years per participant.

After a demographic questionnaire, participants were shown a 1 minute instructional video explaining what Drafty is and how its interface works. This video reviews the same instructions from the welcome screen Drafty shows to first-time unpaid contributors. Then unpaid contributors were given three training tasks, such as "Name a professor who joined their university between the years of 2000 and 2005." Participants performed each interaction type that comprises the user interest profile across various columns to ensure they are familiar with the system.

Normal unpaid contributors are free to use Drafty to explore and edit freely. It was imperative that participants were given the same freedom to create and pursue tasks as normal unpaid contributors. Participants were instructed to create three tasks to perform themselves to simulate a natural inquiry of the data. Examples of tasks they created were:

- *Find my potential supervisor.*
- *Find all the professors at Carnegie Mellon who also obtained PhD at Carnegie Mellon.*
- *Count how many faculty members were hired in the past 10 years by UC Berkeley and Stanford, as well as their research areas.*

The interactions recorded during this part of the validation study were used to build user interest profiles for each participant. After the participant completed all three of their tasks, they answered twelve randomly selected Likert scale questions. The four choices per question were: Not at all interested, Somewhat interested, Interested, or I do not know. Each question asks, ”How Interested are you in...” followed by one of the three following types: a *university name*, *subfield*, or a *professor name from a university*. The content of each question is directly determined by the participant's user interest profile. Each type will be asked four times. There are four methods for selecting the data to ask from the participant's user interest profile. *1) A random data point the participant showed no interest in. 2) Data with the highest interest score per type. 3) A random data point the participant showed some interest in. 4) Data randomly selected from the entire dataset.* This final part of the survey generated 240 total answers that are used to assess the validity of the user interest profile in the following section.

			Mean Interaction Score		
Answer	**N**	**All**	**Prof**	**University**	**Subfield**
Don't know	39	1.13	1.67	1.00	0.00
No Interest	76	0.84	1.47	0.86	0.25
Some Interest	51	1.49	4.18	0.71	0.83
Interest	74	2.73	4.56	1.80	2.54

Table 3.2: Mean interaction score per question for each participant from the validation study.

Validation Results

The weights from Table 3.1 were used to calculate each participant's mean interaction score per question type. The user interest profile is an additive model that relies on accumulating interactions. It works based on the premise that the more interactions the user makes with a specific field, the greater their interest in that field. The results of this study validate the relationship between interactions and interest. Participants' interaction score was, on average 3.24 higher on the data they were interested in versus the data they had no interest in (Table 3.2). A Kruskal-Wallis test was conducted to evaluate the differences among answers from the four Likert scale questions[3]. The test, which was corrected for tied ranks, was significant $\chi_2(2, \text{N} = 240) = 29.6$, $p < 0.001$. This demonstrates that the interaction score from the user interest profile is a significant indicator of an unpaid contributor's level of interest.

3.3 Method

A live experiment was conducted on a publicly-accessible version of Drafty to assess how unpaid contributors behaved in the wild outside of a lab study.

[3] Previous research [6, 50] has shown a Kruskal-Wallis test is appropriate on Likert scale survey questions where group size is unequal

Drafty is a system that requires a large number of unpaid contributors to collect a sufficient amount of data to answer pertinent research questions. So it was shared across various computer science forums such as Hacker News, TheGradCafe, and Reddit CompSci to attract users interested in Computer Science. This contrasts with other systems such as Crowdfill [175] and Soylent [22] that rely on enlisting and compensating crowdworkers. These workers may have varying levels of motivation to use the system and perform tasks. An example title used on Reddit was "Records of 3,600 computer science professors at 70 top universities (US/Canada) help us keep it up to date!." Each post sharing Drafty contained the text: "Wanted to share a computer science resource a couple of us in the Brown University HCI Group has put together. It is a crowd-editable spreadsheet of data from approximately 3,600 computer science professors. For example, where they got their degrees, their subfield of expertise, their join year and rank, etc... It might be useful if you're applying to Ph.D. programs or faculty positions, seeking external collaborators, or to better understand hiring trends in CS departments."

New unpaid contributors are shown a welcome screen on their first visit that includes key information such as "all interactions are captured and used anonymously for studies; double-click a cell to fix a piece of data; and that Drafty is a HCI research system". In addition to the welcome message, Drafty's footer reminds the user that "Drafty is a research project. All interactions are captured and used anonymously for studies." Each new user was randomly assigned to one of three experimental groups. In group 1, unpaid contributors were asked to fix data for professors they showed no interest in. In group 2, unpaid contributors were asked to fix data for professors they did show interest in. Only unpaid contributors assigned to experimental groups 1 and 2 were solicited for data review.

	Number of Attempts			*Mean per Attempt**		*Mean Totals per Visit*		
	Norm.	**Expt.**	**Total**	**Interactions**	**Int.**	**Interactions**	**Int.**	**Visits**
All	1581	1482	3063	11.0	20.5	141.1	290.0	2.7
Not Completed	989	1389	2378	13.7	25.1	134.3	261.4	1.9
Completed	592	93	685	6.7	13.3	151.4	334.3	3.8
Incorrect (Int.)	-	17	17	15.3	64.1	26.8	116.8	1.7
Correct (Int.)	-	24	24	16.2	40.8	27.9	71.1	2.3
Incorrect (Not Int.)	-	4	4	13.5	25.3	71.8	132.8	1.3
Correct (Not Int.)	-	32	32	12.9	21.3	33.1	57.2	2.6
Incorrect (Norm.)	121	-	121	5.9	12.3	30.0	61.3	2.6
Correct (Norm.)	413	-	413	7.0	14.4	37.9	78.0	2.3

Table 3.3: Summary of the correctness of participants' edits to the data. Incorrect/Correct submissions were manually verified by the authors. (*Per Attempt = between each attempt to upkeep data. Int. = Interested, the cumulative score of user interest profile. Norm. = Normal. Expt. = experiment.)

3.4 Results

In the following section, we provide a detailed analysis that investigates the relationships between experimental groups, non-experimental groups, unpaid contributor interactions, and accuracy (Table 3.3). Accuracy is determined by manually checking data submissions using online sources. Unless otherwise stated, non-parametric tests were conducted because the Shapiro-Wilk test of normality and Levene's test showed that the data were not normally distributed and the variances were unequal.

3.4.1 Data Overview

The live experiment, approved by our Human Subjects Protection (IRB) office, ran over a 7 month period ranging from August 25, 2016 to March 25, 2017. During this time unpaid contributors could freely view, edit, and export various academic records using Drafty. Drafty recorded 41,426 interactions from 6,077 unpaid contributors over 7,741 total visits. 809 unpaid contributors had multiple visits at an average of 3 visits per unpaid contributor.

Unpaid contributors submitted data fixes when solicited or self-initiated 31.9% per attempt. Unsolicited unpaid contributors submitted data fixes 37.5% per attempt. When solicited the uninterested group submitted data fixes 5.4% per attempt. The interested group submitted data fixes when solicited 8.8% per attempt. Submission rates can be dependent on the maturity of the dataset. Drafty does not collect an unpaid contributor's personal or demographic information; such as name, age, or gender. Server logs capture IP addresses but are not tied to the user profile nor examined for research.

3.4.2 Active Unpaid-Contributors (more than one visit)

If an unpaid contributor has multiple visits they are active unpaid contributors. It is essential to show how active unpaid contributors make more effective contributions to Drafty, ensuring its long-term viability as a system. Table 3.4 contains a summary of interaction statistics for active versus inactive unpaid contributors. Active unpaid contributors perform 2.4 more interactions per visit than inactive unpaid contributors. They also perform 3.1 more clicks than inactive unpaid contributors. A click can indicate an unpaid contributor has selected a cell to perform additional actions on. For example, to copy the cell contents to their clipboard. They also perform more searches, indicating an active interest in finding specific data. Active unpaid contributors also perform 5 times as many double clicks and create 6 times as many data submissions. These general patterns demonstrate how active unpaid contributors are more engaged and contribute more to the upkeep of a tabular dataset. It is essential to show how active unpaid contributors make more effective contributions to Drafty, ensuring its long-term viability as a system.

Multiple visits can indicate a higher level of interest in the system's data. A one-tailed t-test of unequal variances was performed to compare the number of visits between unpaid contributors who submitted accurate data when solicited vs.

unpaid contributors who submitted inaccurate data when solicited (N = 56, M = 2.5, SD = 7.6) across all experimental groups. Results indicate a significant effect for visits, $t(64) = -1.8, p < 0.05$. By targeting active unpaid contributors, Drafty can make more effective interventions for data review. These results indicate unpaid contributors should be targeted for data review after their first visit.

Unpaid contributors who submit data have approximately 2 times more visits at the time they submit versus those who do not. A Mann-Whitney test was conducted to evaluate the difference in total visits among unpaid contributors who submitted suggestions (M = 4.68, SD = 11.78) against unpaid contributors who did not (M = 2.38, SD = 5.85), for which the difference was significant at $p < 0.001$. This finding coincides with previous results showing that active and engaged unpaid contributors will participate in data review. This demonstrates Drafty's potential to successfully maintain tabular datasets over time.

Interaction Type	**Active**	**Inactive**
Click (Highlight)	3.4	1.1
Double Click	1.0	0.2
Partial Search	2.7	1.8
Complete Search	0.8	0.5
Sort Column	0.3	0.2
Submissions	0.6	0.1
Mean Interactions	8.8	3.7

Table 3.4: Average number of interactions by unpaid contributors per visit segmented by the type of user. "Active" represents unpaid contributors with multiple visits, who perform more interactions on average.

3.4.3 Interactions, Interests, and Edits

Solicited unpaid contributors are visitors to Drafty, that Drafty asks to review data through an intervention. They are part of the interested and uninterested experimental groups. Normal unpaid contributors are those who made submissions for data review using the standard mechanisms within Drafty.

First, we will analyze the interaction habits that generate higher submission rates. Unpaid contributors solicited for data review showed more interesting results than normal unpaid contributors. The following paragraph reviews the relationships between four groups:

- Interested - made submission
- Interested - did not submit
- Uninterested - made submission
- Uninterested - did not submit

A Kruskal-Wallis test was conducted to evaluate differences among four conditions/groups when unpaid contributors were solicited to review data on the mean number of interactions and the mean score of interactions an unpaid contributor performed before being solicited to find missing data.

Unpaid contributors are more likely to find and submit missing data when they make more interactions and have higher interest scores between solicitations. They also make more total interactions per visit. The number of interactions between solicitations is significantly different $\chi_2(3, \text{N} = 93) = 24.5$, $p < 0.001$. The cumulative interest score of interactions in between solicitations is significant $\chi_2(3, \text{N} = 93) = 17.8$, $p < 0.001$. Finally, the total number of interactions per visit is significantly different $\chi_2(3, \text{N} = 93) = 38.6$, $p < 0.001$. This indicates unpaid contributors who are alerted too early could potentially have a negative reaction to the pop-up window used in solicitations. Soliciting reviews of data should be done after the unpaid contributor has made a certain number of interactions to generate a robust user interest profile. Also, the total cumulative interest score is significant $\chi_2(3, \text{N} = 93) = 30.5$, $p < 0.001$. Follow-up tests were conducted to evaluate pairwise differences among the four groups. A Bonferroni correction was applied to control for Type I error. It showed no significant difference between the four groups.

Generally, the more interactions per session, the greater the chance an unpaid contributor will edit data when solicited. This result mirrors a similar study focusing on tagging data [242].

Condition	**Precision**	**Recall**
Uninterested group	57.1%	3.0%
Interested group	88.9%	9.2%
Normal unsolicited	75.8%	28.4%

Table 3.5: Unpaid contributors asked to review data they are interested in have higher precision accuracy on data submissions normal unsolicited and uninterested unpaid contributors.

While submission rates are useful to show an engaged unpaid contributor population, accuracy is a better metric to understand Drafty's ability to maintain up-to-date tabular data. We use accuracy metrics, precision and recall, derived from the information retrieval community. Precision is the number of correct submissions over the number of submissions per group. Recall is the number of correct submissions over the number of solicitations per group. Refer to Table 3.5 for a summary of precision and recall per data review condition.

Unpaid contributors who were solicited to fix data they are interested in are three times more likely to submit accurate data than unpaid contributors who are asked to fix data they are uninterested in. The highest precision is achieved when an unpaid contributor is solicited to fix data based on their interests. This demonstrates Drafty uses a more effective method for data collection and verification than traditional crowdsourcing methods such as those found in CrowdFill [175] and [173]. In addition to these findings, unpaid contributors submitted 81 fixes for subfield area of expertise. After manual verification, 92.6% of the submissions for subfield were deemed accurate. This is a substantial increase in accuracy for fields requiring domain-specific knowledge compared to traditional methods [173].

Solicited and normal unpaid contributors who made accurate submissions had more interactions per visit. This indicates the more engaged the unpaid contributor, the better they are at maintaining a tabular dataset. A one-tailed t-test of unequal variances compares the number of total interactions between unpaid contributors who submitted accurate data when solicited (N = 21, M = 65.6, SD = 58.5) vs. unpaid contributors who submitted inaccurate data when solicited (N = 56, M = 36.4, SD = 39.9). The test determined that there was a significant difference in the total interactions per visit, $t(27) = 2.1, p < 0.05$. The same type of t-test compared unpaid contributors who submitted accurate data when not solicited (N = 121, M = 30.0, SD = 23.2) vs. unpaid contributors who submitted inaccurate data when not solicited (N = 413, M = 37.9, SD = 38.8) across all experimental groups. Results indicate a significant effect for the number of total interaction scores per visit, $t(333) = -2.8, p < 0.01$.

Interaction Type	**Incorrect**	**Mixed**	**Correct**	**All**
Click (Highlight)	7.0	5.8	6.1	6.2
Double Click	1.8	2.3	2.0	2.1
Partial Search	3.6	2.6	7.0	5.1
Complete Search	1.1	0.9	1.6	1.3
Sort Column	0.6	0.5	0.4	0.4
Submissions	3.2	2.6	0.4	2.5
Mean Interactions	17.5	14.7	19.6	17.8

Table 3.6: Mean number of interactions by unpaid contributors per visit, separated by accuracy. (Incorrect = unpaid contributors who only made inaccurate submissions. Mixed = unpaid contributors who made accurate and inaccurate submissions. Correct = unpaid contributors who only made accurate submissions.)

Unpaid contributors who only submitted accurate data across all conditions perform more partial and complete searches than other unpaid contributors (Table 3.6). This matches earlier observations showing active unpaid contributors perform more partial and complete searches than inactive unpaid contributors. Both findings support the decision to give higher weight to searches over clicks within the user interest profile.

3.5 Discussion

This section reviews fundamental questions and observations about developing and using a public data system like Drafty to enlist crowd contributions to upkeep tabular data. Drafty allows for tabular data maintenance deploying both micro- [175] and macro- [86] task-based approaches. By combining the strengths of these systems with a validated user interest profile model, Drafty can maintain tabular datasets over long periods at a lower cost and higher efficiency than traditional crowd-powered systems. Current results show that interest-based profiling increases data accuracy by soliciting unpaid contributors to find missing data they are interested in. Results also show Drafty is highly effective at sustaining data quality for fields that require domain-specific knowledge, such as subfield area of expertise.

Results demonstrated that unpaid contributors with more visits to Drafty were more engaged. They submitted more fixes and had different interaction patterns and habits. In the future, finding the most effective unpaid contributors is essential. Drafty could provide better interventions fewer times and achieve more accurate results. Attracting and keeping its base of successful unpaid contributors is similar to other systems employing various methods to build trusted groups of paid crowdworkers to ensure repeated efforts of high-quality contributions [22].

Drafty can prioritize conflicted data for solicitation and upkeep. This will allow Drafty to scale its effectiveness for mature datasets, allowing the feedback loop to better verify new data suggestions. For example, Drafty can prioritize cells with multiple suggestions whose standard deviation of confidence scores is within a certain threshold. This allows Drafty to identify uncertain data and make solicitations to fix it preemptively. Data verification is an open research area in crowdsourcing.

3.6 Conclusion

This chapter reports the design, implementation, and evaluation of Drafty, a public data system that recruits contributors from the users of tabular data. Drafty's unpaid contributors fix tabular data through solicited and unsolicited methods. Their interactions were captured and analyzed to build user interest profiles, which were validated by a survey afterward. In a longitudinal experiment in the wild, the interest profiles were used to solicit unpaid contributors to fix data they were either interested in or had not shown interest in based on their interactions.

Unpaid contributors asked to fix data they are interested in are more than three times more likely to submit accurate data than unpaid contributors asked to fix data they are uninterested in. Unpaid contributors who performed more interactions in between being prompted to fix data were not only more likely to submit data, but their submissions had higher levels of accuracy. Successful unpaid contributors often actively searched for data and would perform more search actions to find tabular data to review. This experiment has shown that using user interest profiles help improve the accuracy of tabular data and helps build a self-sustaining dataset. Overall, this chapter is the first evidence in the dissertation where users become custodians, and outdated data becomes an obsolete concept.

Chapter 4

Developing Paid Crowd-based Verification Strategies to Collect-Verify Tabular Data

This chapter presents recommendations for creating new data verification strategies by observing real-world novice requesters collect and verify a tabular dataset of tenure-track Computer Science faculty academic profiles. This chapter is a summarized version of an unpublished study that serves as the motivation for the case studies presented in chapter 5. For this research, I acted as the first author and was responsible for running the user study, part of the analysis, and the majority of the writing.

4.1 Introduction

Tabular datasets, structured as unique rows and corresponding columns, are used by governments, businesses, and scientists to organize information and maintain knowledge [40]. Building tabular datasets from human knowledge and online information sources is a collective information retrieval and organization task [239]. These tasks easily translate to small repeatable micro-tasks appropriate for many paid crowdsourcing scenarios. While paid crowdsourcing offers an initial step to easily collect information, there is a lack of research on verifying and improving the accuracy of crowd-collected tabular data. Error-prone tabular data [172] can lead to dire consequences, like the austerity measures imposed on Greece after the 2008 financial crisis [93]. In this age of information overload, inexperienced requesters' ability to improve rapidly changing tabular datasets' accuracy is vital to enable continuous positive impacts.

This study focuses on how three data verification strategies, *Find and Fix*, *Majority Rule*, and *Expert Rule*, all developed by novice requesters in a naturalistic scenario, can improve the accuracy of tabular data. We describe a study where we observe novice requesters run data collection and verification tasks to create a dataset of the academic development of over 3,500 faculty from Computer Science departments in the United States and Canada. This realistic task is difficult because the information is spread across department websites, faculty curriculum vitae, bios, and other sources. It also poses challenges for crowdworkers lacking domain-specific knowledge in computer science or academia, e.g., terms like Sc.B. might be ambiguous, or machine learning and computer education might seem semantically similar.

How should requesters, inexperienced with crowdsourcing, create tasks, set payments, and manage crowdworkers for data verification tasks? This work seeks to provide recommendations to answer these questions to help current and future requesters. In this work, we define paid verification as using crowdsourcing to systematically verify and improve the accuracy of data previously collected by paid crowdworkers.

This chapter focuses on quantitative analysis and qualitative interpretations of novice requesters' reports to make one primary contribution: we report the effectiveness of different crowd-based data verification strategies to improve data quality and how factors such as crowdworker qualifications, compensation, and bonuses influence data quality. A secondary contribution is a discussion of initial recommendations for requesters performing data verification.

4.2 Method

To promote a naturalistic approach and avoid biases due to our experience with crowdsourcing, we did not conduct the verification strategies ourselves. Instead, similar to Papoutsaki et al. [173], twenty undergraduate and graduate students from a human-computer interaction seminar acted as requesters and employed crowdworkers to collect and verify tabular data as part of a graded class assignment. Students read three seminal works [21, 68, 122] and watched a recorded talk by Adam Marcus on working with crowdworkers [151]. Students were given an assignment to create a dataset of complete academic profiles of computer science faculty from 80 top computer science programs in the United States and Canada, according to the US News ranking[1]. Each faculty profile consists of their affiliated university, the year they joined as faculty, professor rank, subfield area of expertise, where they received their Bachelors, Masters, and Ph.D., gender, and links to their profile photo and information sources. Each student acted as a requester to collect and verify data from four unique universities over two and a half weeks.

We chose a similar experimental design and dataset with [173] to control for factors besides crowd-based data verification, as the students created similar tasks with similar constraints during the same period of time. This experimental design allows us to investigate the patterns of multiple similar crowd requesters aiming to achieve the same goal with the same resources.

Each student completed the assignment in three phases: testing, data collection, and data verification. They received $40 in Amazon credit for use on Amazon Mechanical Turk. They were required to spend $8 in total ($2 per university)

[1] http://www.usnews.com/best-graduate-schools

for testing. During testing, they could experiment with Amazon Mechanical Turk and various strategies. The remaining $32 ($8 per university) were split equally between data collection and verification.

This class-based approach enables a unique understanding of *cohorts of people crowdsourcing with the same overall objective*, allowing for comparisons and qualitative observations. Our Human Subjects Office determined these procedures to be exempt from IRB review. Following established human subject guidelines, all students acquired written consent from the Amazon Mechanical Turk crowdworkers they employed as part of the task.

During the *data collection phase*, we randomly assigned students one of the five crowd-based data collection strategies: Brute-Force, Column-by-Column, Iterative, Classic Micro, or Chunks as identified by Papoutsaki et al. [173]. Controlling for collection strategies allows us to study requesters' habits, decisions, and strategies during the verification phase. While not a focus of this work, our results reproduce the findings from Papoutsaki et al. [173] in a controlled setting.

4.2.1 Crowd-Based Data Verification Phase

Students could design and implement one or a combination of their crowd-based verification strategies per assigned university. We observed that they primarily used one dominant verification strategy. To simulate real-world crowdsourcing scenarios, students could run verification tasks during the data collection phase. This procedure involves non-trivial data verification tasks with actual practical utility. These realistic tasks require workers to correctly find and interpret subjective information sometimes requiring domain-specific knowledge of computer science. Time and financial constraints simulate a realistic environment, where requesters and crowdworkers create a temporary ecosystem of competing tasks and agendas. The students can be seen as novice requesters (due to their lack of experience) with realistic motivators of time and grades. After 8 days of testing and 16 days of data collection and verification, students collected approximately 3,500 faculty records and wrote detailed individual project reports ranging from 10 to 30 pages.

4.2.2 Crowd-Based Data Verification Strategies

We identified three crowd-based data verification strategies from the student journals using grounded theory methodologies [211]. These strategies share similarities to those reported in literature. We speculate that once the responsibility of developing a data collection scheme is removed, novice requesters converge faster to a smaller set of verification strategies employed by advanced requesters. We describe below the strategies that were designed by the students and the effectiveness of their choices.

Find and Fix acts as a feedback loop between data verification and collection, with one or more crowdworkers assigned to review data and correct errors. Similar to the Find-Fix-Verify scheme [21], a crowdworker can fix data if they consider them incorrect. This strategy appeared in various forms in the study. In some cases, crowdworkers would review and fix a single piece of information provided by the native interface of Amazon Mechanical Turk or review a spreadsheet of the entire dataset where crowdworkers could edit preexisting data and fill in missing fields. This strategy allows crowdworkers to verify if a blank cell should remain blank (e.g., faculty who do not have a master's degree). A requester chooses when to consider the data to be correct, as there is no obvious endpoint for this verification strategy.

In ***Majority Rule***, multiple crowdworkers collect data for the same entity, creating duplicates for each data point. The most common response per data point is deemed correct. In this strategy, an odd number of duplicates is preferable to simplify the process. Various quantitative metrics can break ties (e.g., crowdworker speed, crowdworker qualifications, previous experience with particular crowdworkers, or the crowdworker's trustworthiness). Similar to Majority Decision [97], the speed of *Majority Rule* is one of its benefits: it can quickly scale over large datasets with the use of automated programs to determine the final result from the duplicates. The main drawback of *Majority Rule* is that it requires redundant data points, increasing the overall cost and time needed to generate duplicates.

In ***Expert Rule***, crowdworkers also collect multiple entries for each data point. *Expert Rule*'s key difference with *Majority Rule* is that it requires a crowdsourced vote to break ties. During the verification stage, requesters recruit crowdworkers to choose the correct data point. Requesters usually employed two crowdworkers per disagreement. Verification is complete if both crowdworkers pick the same entry. If there is a dispute, then a third crowdworker is recruited to resolve it. The third crowdworker is often recruited using custom qualifications or has an established relationship with the requester. Their vote is considered the correct choice and completes the verification step. *Expert Rule* allows the requester to leverage a large pool of crowdworkers to narrow down the initial duplicate entries at a low cost. This strategy gives requesters an obvious stopping point for the verification process.

4.3 Results

In this section, we investigate the relationships among data verification strategies, qualifications, payments, and accuracy. The majority of our quantitative analysis uses our relatively accurate *ground truth* dataset. We use two *measures of accuracy* to compare the faculty dataset from this study to the *ground truth* dataset. Borrowing the concepts of recall and precision from information retrieval, we define the following two metrics for each university. *Recall* is the ratio of correctly identified fields across all faculty records that were expected. If a field is missing, it is considered incorrect. We ignore entries for professors who were not in the *ground truth* dataset. *Precision* is the ratio of correctly identified fields within the faculty records that a student collected for a specific university. All data analyzed using MANOVA passed the Shapiro-Wilk test of normality. MANOVA and t-tests use the Levene's test of homogeneity of variances, unless otherwise specified. All post-hoc analyses following MANOVA were conducted using Tukey's procedure, which corrects for family-wise error-rate.

Dimension	Value	N	Recall	Precision
Qualification	master	12	0.64	0.75
	preset	28	0.57	0.73
	custom	22	0.63	0.76
Bonus	offered	32	0.63	0.77
	none	15	0.55	0.70
Communication	passive	7	0.63	0.78
	business	11	0.55	0.73
	selective	5	0.56	0.76
	rapport	6	0.63	0.77
	none	18	0.64	0.73

Table 4.1: The effect of different qualifications and bonuses on mean recall and precision, which were computed per university. N = a total number of universities per dimension. Requesters were twice as likely to offer bonuses per university during verification tasks.

4.3.1 Task Dimensions Describing Data Verification Strategies

Applying open coding on student reports, we identified two unique dimensions that describe features of crowd-based verification strategies, as shown in Table 4.1: crowdworker qualifications and compensation. To the best of our knowledge, this is the first attempt to investigate the dimensions of verification strategies chosen by novice requesters.

Qualification

Three types of qualifications were used to assess the eligibility of crowdworkers per task. Amazon Mechanical Turk automatically assigns **Master Qualification** once a crowdworker has completed enough tasks while maintaining high accuracy. A potential issue with Master crowdworkers is they might lack domain-specific knowledge for specialized tasks. **Preset Qualifications** allow the requester to specify multiple predefined performance criteria (e.g., crowdworkers with at least 500 completed tasks and a minimum approval rate of 95%). These flexible values allow access to a larger pool of crowdworkers than Masters. **Custom Qualifications** are manually created and assigned to crowdworkers by the requester. They can be used to select a group of high-performing crowdworkers.

Payment per Task and Bonuses

Crowdsourcing platforms such as Amazon Mechanical Turk compensate crowdworkers for their effort, but often do not propose guidelines for fair compensation. We considered three measurements of total **payments per task**: the total pay, the pay during the data collection phase, and the pay during the data verification phase. These payments were normalized within a university by the total number of collected professors. Requesters can use **bonuses** as additional monetary compensation beyond the typical payment for completing a task. Students offered bonuses for different reasons. For example, crowdworkers received bonuses for completing an exceptional amount of work or if they completed a task quickly.

	Brute-Force	Column-by-Column	Iterative	Classic Micro	Chunks
N	6	12	12	7	10
Recall	54.8%	63.0%	52.3%	69.1%	65.1%
Precision	70.3%	73.0%	76.0%	75.5%	76.8%

Table 4.2: Recall & precision per data collection strategy.

Communication with crowdworkers

We coded five levels of communication determined through independent coding and solving inconsistencies through discussions among the authors. The first level is **none**, with no communication between the requester and the crowdworker. The second level is **passive**, which involves minimal communication with the crowdworker, e.g., when the requester only replies to emails about rejection reconsideration. Scenarios where the requester had more active communication with the crowdworker, e.g., to thank crowdworkers for good work, are classified as **business**. When the requester established rapport with some of the crowdworkers, e.g., those with the best performance or those that had special qualifications, we refer to it as **selective rapport**. Finally, **rapport** refers to cases where the requester established strong relationships with crowdworkers, e.g., a phone conversation to discuss tasks.

4.3.2 Accuracy across Data Collection Strategies

Over half of the students conducted part of their verification during the data collection phase. A two-way MANOVA tested the effect of the collection strategy together with the verification strategy on total and per column recall and precision. The results did not show an interaction effect between collection and verification strategy.

Table 4.2 shows the average recall and precision for each data collection strategy. Classic Micro achieved the highest recall and relatively high precision. Overall, our results reproduce the findings from [173]. While Iterative, Classic Micro, and Chunks had the highest precision when including corrections across all data types, the collection strategy did not affect recall or precision. However, collection strategy has a significant effect on the recall ($p = 0.002, F_{(4,34)} = 5.154$) and precision ($p = 0.025, F_{(4,34)} = 3.194$) of subfield. Column-by-Column had the highest recall and precision for subfield. In this strategy, crowdworkers would repeatedly collect only one data type, or column, thus enabling them to gain the domain-specific knowledge required to correctly interpret a professor's subfield.

4.3.3 Accuracy across Data Verification Strategies and Task Dimensions

Verification Strategy and Accuracy

To assess the recall and precision of verification strategies across all columns and per column, we performed a two-way MANOVA assessing the effect of the verification strategy. Figure 4.1 (a) shows the mean recall per data verification

	N	Subfield	Join Year	Rank	Masters	Bacherlors	Doctorate
Find and Fix	*17*	45.7%	52.9%	82.4%	58.6%	62.6%	75.0%
Majority Rule	*13*	35.2%	38.8%	71.4%	45.0%	54.4%	62.2%
Expert Rule	*17*	37.2%	57.6%	69.9%	48.7%	55.7%	68.6%

Table 4.3: Mean recall achieved per data verification strategy per columns. N = total number of universities per strategy. Subfield has the lowest recall and requires domain-specific knowledge for crowdworkers to interpret a professor's research area.

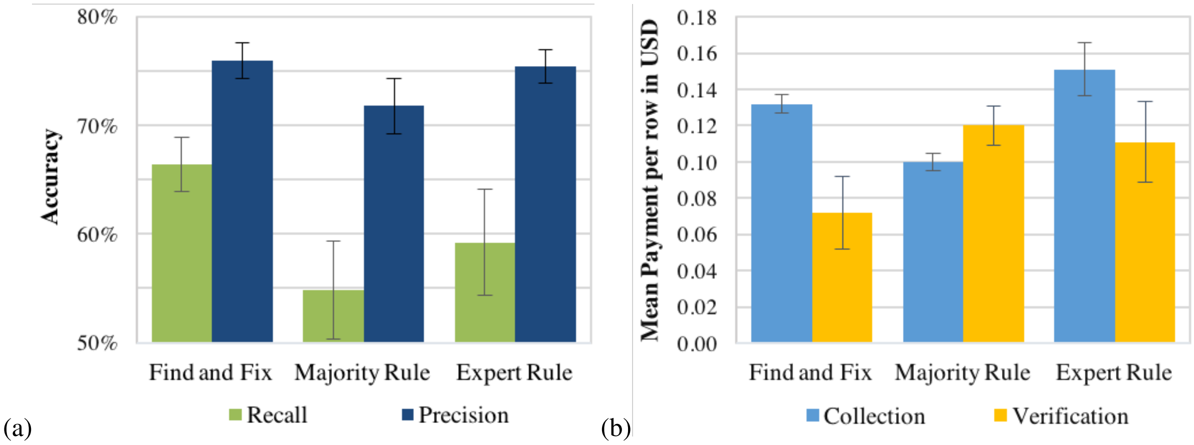


Figure 4.1: (a) Mean recall and precision per data verification strategy. Error bars represent standard errors. (b) Mean payment per row on collection and verification tasks per data verification strategy. Error bars represent standard errors.

strategy, split across six columns. Results of a two-way MANOVA show the choice of verification strategy has a borderline significant effect on the recall of a professor's subfield ($F_{(2,34)} = 3.240, p = 0.052$). *Find and Fix*'s recall for subfield is 9.4% higher than *Expert Rule* and 11.2% higher than *Majority Rule*. *Find and Fix* might excel at fields requiring domain-specific expertise, like subfield, because it allows requesters to re-release tasks to verify and clean single cells. An explanation for the lower recall and precision for *Majority Rule* and *Expert Rule* is their duplicate collection efforts do not allow crowdworkers to see previously submitted values. Thus, they do not allow crowdworkers to learn the subtle differences in fields requiring domain-specific knowledge. For example, crowdworkers commonly mislabeled a professor's subfield as Computer Education, maybe because they teach Computer Science.

For Majority Rule, we found that the recall of Masters information is significantly higher ($F_{(1,39)} = 4.246, p = 0.045$) when Majority Rule is not used. Similarly, the precision of Masters information is also higher when Majority Rule is not used, though the difference is borderline significant ($F_{(1,39)} = 3.671, p = 0.062$). The difference in the precision of rank information is also borderline significant ($F_{(1,39)} = 3.696, p = 0.061$) with higher accuracy when Majority Rule is not used.

Qualification and Accuracy

Students could mix different qualifications per task. Among the 47 universities, 25.5% were collected by Master crowdworkers, 59.5% with Preset, and 46.8% with Custom Qualification. Many students experimented with Master Qualifications while testing and reported trouble recruiting crowdworkers due to requirements for higher compensation. This made Preset Qualifications a more popular choice. A two-tailed t-test was used to compare the mean accuracy of data collected with and without each Qualification level (Table 4.1). Using Master or Custom Qualifications showed no significant effect. The absence of Preset Qualifications had a significant effect on overall recall ($t(45) = 2.139, p = 0.0379$) and precision ($t(45) = 2.305, p = 0.0258$). In both cases, the accuracy was lower when Preset Qualifications were used by 10.07% and 4.8%, respectively. Two individual data types have lower recall when Preset Qualifications were used: subfield ($t(45) = 2.468, p = 0.0175$) and doctoral degree ($t(45) = 2.183, p = 0.0343$). This might have occurred because no Qualification level targets crowdworkers with domain-specific knowledge. Previous work has shown that crowdworkers with expertise on specialized tasks perform better [202, 225]. The qualifications available did not allow participants to target crowdworkers with domain-specific knowledge. Given this constraint, perhaps Preset Qualifications is not ideal for domain-specific tasks.

Communication with Paid Crowdworkers and Accuracy

We performed a one-way ANOVA to test the communication levels between requesters and crowdworkers and data accuracy. We found no significant effects. A few explanations emerged from the student journals. Some students had high confidence in their initial data collection and felt no need for communication. Several students used communication to recruit trusted crowdworkers to perform specific verification tasks. Students who used the *Expert Rule* strategy used trusted crowdworkers to cast tie-breaking votes. Also, students reported that communication with crowdworkers was a useful method to gain feedback on payment tasks and payment methods. Notably, there were several examples where students modified payment per task after communicating with a crowdworker.

4.3.4 Analysis of Cost per Contribution

This section discusses findings on the effects of base payments and bonuses. We define base payment as the average dollar amount spent per professor; including payments for data collection and verification. Similarly, we computed the money spent on bonuses per professor.

Payments and Column Difficulty

To investigate the effect of payment amounts on accuracy across columns, we performed a multivariate multiple linear regression. The model includes three independent variables: data collection payment, verification payment, and bonus payment (all per professor). Dependent variables include overall and per column recall and precision. Values of condition indices and variance proportions of the regression model indicate no co-linearity issues among the three independent variables.

Students often used bonuses at the end of the data collection and verification phases to find missing or verify conflicted data. Bonuses have a significant correlation with overall precision ($\beta = 0.349, p = 0.017, R^2 = 0.214$). Students using bonuses achieved a 76.77% mean precision, 6.84% higher than those who did not. Certain columns benefit from higher collection or verification payments. Subfield is the most subjective data type as it requires domain-specific knowledge. For subfield, both collection ($\beta = 0.332, p = 0.023$) and verification payment ($\beta = 0.333, p = 0.016$) are significant predictors of its precision ($R^2 = 0.271$). Its recall increases with higher collection payment ($\beta = 0.374, p = 0.014, R^2 = 0.212$).

Collecting the Bachelors and Masters degree is challenging because professors tend to omit them from their websites. Thus requiring crowdworkers to search through multiple information sources. We observe that bonus payments are a motivational factor to find elusive data. Higher bonuses are associated with higher precision for Bachelors ($\beta = 0.435, p = 0.003, R^2 = 0.269$). Bonuses are also a significant predictor of both recall ($\beta = 0.330, p = 0.039, R^2 = 0.104$) and precision ($\beta = 0.331, p = 0.04, R^2 = 0.097$) for Masters.

Figure 4.1 (b) depicts the mean payment per professor per data verification strategy for the collection and verification phases. This agrees with work that shows after filtering, higher-demand tasks lead to higher quality data [114].

There was a high variance among rewards. Students stated that time constraints and task difficulty were factors for their budget allocation. Some students experimented early to find the balance between minimum reward and maximum accuracy. Students reported receiving comments from crowdworkers who suggested appropriate rewards. Releasing similar tasks increased the assignment's difficulty as students now had to compete with their classmates—workers communicated with the students over the variations in payment and rewards for these similar competing tasks.

Payments per Verification Strategy

Qualitative observations from the above analyses suggest the effectiveness of verification payments may vary depending on the verification strategy used. Therefore, we conducted further analyses using each verification strategy on subsets of data verified. Figure 4.1 (b) indicates students spent more per professor in *Majority Rule* than other verification strategies. Also, Table 4.3 shows that *Majority Rule* had the lowest recall for each data type (column) except rank. We first performed a multiple linear regression with verification payment as the independent variable and overall and per column precision and recall as the dependent variables. No significant effect was found when using *Find and Fix* or *Expert Rule*. In *Majority Rule*, verification payment is found to be a significant predictor for both precision ($\beta = 0.783, p = 0.002, R^2 = 0.614$) and recall ($\beta = 0.791, p = 0.001, R^2 = 0.625$) of Bachelors information, as well as for both precision ($\beta = 0.587, p = 0.035, R^2 = 0.345$) and recall ($\beta = 0.642, p = 0.018, R^2 = 0.412$) of subfield information.

Student journals support this insight. Students released more tasks to verify missing and erroneous data, such as Bachelors and subfield than other types. The students, already familiar with computer science, could recognize errors in subfield data. Additionally, professors typically have bachelor's degrees. Thus it is safe to assume a blank entry is incorrect. *Find and Fix* had the lowest cost per professor, most likely because students could release smaller, cheaper tasks focused on single data points instead of collecting duplicates or hiring trusted crowdworkers to break ties.

Pay Level Analysis

We performed a one-way ANOVA to test the effect of pay level on total and per column data accuracy. We considered pay level during data collection and data verification separately and performed separate ANOVA for each.

We found that pay level for data collection tasks has no significant effect on total recall or precision. However, it has significant or borderline significant effects on recall and precision of the professor's Bachelors and Masters degree information. For Bachelors information, data collection pay level has borderline significant effect ($F_{(2,44)} = 2.817, p = 0.071$) on recall and significant effect on precision ($F_{(2,44)} = 5.59, p = 0.007$). Post-hoc analyses show that Medium pay level performed best for both recall and precision, and led to a significantly higher (14.37%) precision than Low pay level ($p = 0.007$). For Masters information, data collection pay level has significant effect on both recall ($F_{(2,44)} = 4.065, p = 0.024$) and precision ($F_{(2,44)} = 3.935, p = 0.027$). Same as Bachelors information, Medium pay level had significantly higher accuracy than Low pay level in both recall (by 16.23%, $p = 0.03$) and precision (by 15.73%, $p = 0.021$).

The results show that offering higher payments for data collection tasks does not seem to effectively increase overall recall and precision. However, using medium standard payment leads to better accuracy for Bachelors and Masters information than offering low payment. One possible explanation for this is that Bachelors and Masters information is relatively more difficult to find and with lower payment crowdworkers may be more inclined to do minimal work and not put much effort. Medium pay offers more motivation for crowdworkers to do a better job. A high pay level, on the other hand, might mean that the requester has to make some trade-off elsewhere, given the fixed total budget.

Offering higher standard payment on verification tasks tends to lead to more accurate results. Pay level during data verification has a significant effect on data precision ($F_{(2,44)} = 3.558, p = 0.037$). Post-hoc analysis shows that precision is higher by 6.49% under High pay level compared to Low pay level.

4.4 Recommendations for Novice Requesters

We analyzed students' habits and strategies to investigate their effect on crowdsourcing tasks and results. We summarize our findings into data verification recommendations for requesters new to crowdsourcing.

4.4.1 Budget for Collection and Verification Tasks

We observed complex relationships between payment allocation, data quality, and verification strategies. We found that spending more on collection or verification led to better results for difficult-to-collect data, e.g., subfield and Masters. Also, our results show that increasing verification payment significantly affected the accuracy of selective columns only when using Majority Rule.

A requester should consider a task's difficulty, time to complete, and payment per verification strategies. Our observations agree with past research that shows that larger budgets do not necessarily increase quality. A higher budget can lead to lower accuracy due to scalability issues [230]. Given a fixed budget, requesters should experiment with different payment allocation schemes while considering task difficulty. Our results suggest that bonus payments are helpful for error-prone tasks and limit extraneous verification tasks. Whether it is worth allocating more budget to verification could depend on task difficulty.

Novice requesters are often unsure of what fair compensation is per task. Some platforms, such as Dynamo [196], recommend a minimum hourly wage, and crowdworkers expect this level of payment. In our study, students reported that crowdworkers requested that payments should match the US minimum wage. Small test tasks can help to identify the appropriate payment based on task size and difficulty. This can help requesters scale their verification strategies to produce larger datasets within a budget.

4.4.2 Select the Best Input Methods per Data Type for Crowdworkers

The lack of verification on free text input often led to additional verification tasks. Partially restricted inputs were used for fields that required university names to allow crowdworkers to pre-select from a verified list of university names or input new values. Subfield, which required domain-specific knowledge, was the most common restricted input.

Requesters have three input options for crowdworkers when designing crowdsourcing tasks: i) free-text input, ii) partially restricted input, or iii) restricted user input to a discrete set of options. Free-text input is simple but requires additional review to remove typos and merge variations. This noise can be removed through additional verification and cleaning tasks. A benefit of free-text input is that it dissuades cheating [122]. Partially restricted inputs are a hybrid approach. Paid crowdworkers can select from a predefined list, similar to auto-complete functionality, or add a new piece of data. For example, universities have nicknames, but predicting all possible university names would require substantial upkeep. Providing a drop-down list with preset values can eliminate noise in user input caused by typos and variations in naming conventions. However, this discrete set of options makes it easier for crowdworkers to cheat and quickly choose a random option. Restricted inputs can be coupled with bonuses to demotivate cheating.

4.4.3 Choosing the Right Strategy to Verify Domain-Specific Data

Our results demonstrate that the effectiveness of strategies differs across data types. For example, the subfield requires domain-specific knowledge of computer science, whereas Doctorate does not. Our observations suggest that a crowdworker can gain domain-specific knowledge by completing a relevant task. For example, *Find and Fix* potentially exposes crowdworkers to domain-specific data. Also, *Expert Rule* allows requesters to re-recruit a trusted crowdworker to break ties, thus building on the recommendation of Daniel et al. [57]. These repeat crowdworkers are usually qualified to assess disparities in data requiring domain-specific knowledge. Other work has pre-screened crowdworkers to match their skills and expertise to specific tasks [159]. In contrast, verification strategies such as *Majority Rule* suffer from over-automation; it is easy to collapse, and correct data, but a complicated answer may get lost among various duplicates. In summary, two wrongs may not always be correct. However, for easier-to-find data, such as Doctorate, *Majority Rule* could be sufficient, thus providing a simple, low-cost strategy to verify data quickly.

4.4.4 Recruiting the Right Paid Crowdworker for the Right Task

Previous research shows little difference between expert and non-expert crowdworkers for routine tasks. This relationship changes for specialized tasks [202, 225]. Anecdotal evidence from student reports further supports the following idea: if students had employed crowdworkers with domain knowledge in computer science or academia, they would acquire data with higher accuracy. Crowds with specific knowledge [169] are better at completing knowledge-intensive tasks [62]. Amazon Mechanical Turk's built-in qualifications (e.g., Master or minimum approval rates) did not significantly increase and, in some cases, decreased the accuracy of specialized columns. These basic Preset Qualifications provide little information about a crowdworker's familiarity with specific subjects. They might exclude crowdworkers who possess specialized knowledge relevant to the task. Therefore, creating custom tasks to identify crowdworkers with the desired expertise may be worthwhile to recruit them to verify specialized data may be worthwhile.

4.4.5 Detect Malicious Crowdworkers to Improve Accuracy

We observed several malicious crowdworkers sabotage data while benevolent crowdworkers could save it. Some requesters reported mal-intent when interacting with crowdworkers, while others reportedly maintained good relationships to the point that receiving payment was secondary to the task. Other requesters were praised on social crowdworker forums. Our findings from both studies emphasize the effects of communication, recognition, and payment: crowdworkers appreciate requesters that recognize their effort and treat them as individuals. Requesters should consider framing their language to the crowdworker in an accommodating and understanding manner. While crowdworkers and requesters do not physically communicate through the interface, one must remember their voice and expression during communications and interactions. Novice requesters can be unaware of the various channels of communication that exist between crowdworkers, such as forums and websites [12]. The crowdworker community assesses requesters on their communication, generosity, fairness of payment, and promptness, using this collective feedback to identify preferred versus unfavorable requesters. For novice requesters, it is essential to establish a good reputation, as this could affect their ability to attract crowdworkers in the future.

4.5 Conclusion

This chapter analyzes the accuracy and costs of paid verification strategies for tabular data and recommendations to aid future novice requesters. Our classroom-based study shows how different data verification strategies can minimize costs and increase the accuracy of a tabular dataset. Results show that increased payment per task does not directly increase accuracy. However, the possibility of earning extra money is a positive factor that can increase the accuracy of edits. Our findings demonstrate that some design choices made by novice requesters are consistent with the best practices recommended in the community.

The task of building and verifying a dataset has critical elements of real-world importance, subjectivity, and ease of discovery. It is difficult to match a novice requester's exact scenario when verifying a tabular dataset to substantiate generalizable claims. While the approximation we used is students learning crowdsourcing, they likely have differences in demographics and motivation. This work focuses on one type of tabular data, computer science professors, to investigate verification strategies. Different data could present different challenges not covered in this work. Future work could focus on controlled studies to evaluate the verification strategies identified in this work.

We increasingly rely on tabular datasets' accuracy and domain-specific knowledge to make critical decisions. As society learns to cope with an ever-growing amount of information, inexperienced requesters' ability to verify and improve the accuracy of tabular data grows in importance. Remote work environments will create naturalistic scenarios similar to what the students in this study experienced. The strategies and observations discussed in this chapter provide a starting point for requesters creating data verification tasks and serve as a foundation for the case study on paid verification strategies in chapter 5 of this dissertation.

Chapter 5

Case Studies on the Motivation and Performance of Paid and Unpaid Contributors Who Verify and Maintain Evolving Tabular Data

This chapter presents two case studies evaluating paid and unpaid crowd contributions to evolving tabular datasets of tenure-track Computer Science faculty academic profiles. This chapter is a summarized version of [224], where I was the first author and responsible for all aspects of the user studies, system design and implementation, analysis, and the majority of the writing. This chapter also contains new results and discussions on the trade-offs between the cost or time to wait for a contribution within paid and unpaid strategies; see Section 5.5. As part of the contributions of this chapter, the associated datasets and labels from this work are publicly available[1].

5.1 Introduction

This chapter features two case studies that focus on the motivation and performance of contributors during the maturity phase of a tabular dataset's life cycle compared to the large body of related work focusing on improving unstructured data [117, 209]. Tabular datasets, structured as unique rows and corresponding columns, enable users to engage in structured information exploration and retrieval to augment semantic knowledge bases. [239]. Disseminating tabular

[1] Available at http://drafty.cs.brown.edu.

information from human knowledge and online information sources is a collective information retrieval and organization task often powered by peer production systems and strategies. However, accurate datasets are difficult to maintain during the maturation phase. Error-prone tabular data [172] can lead to dire consequences, like the austerity measures imposed on Greece after the 2008 financial crisis, which were based on a spreadsheet with numerous errors that inaccurately represented the relationship between public debt and GDP growth [93]. It is imperative to verify and maintain tabular datasets' accuracy to enable positive impacts continuously. Not all data is static; to maintain its accuracy as it grows in size and evolves, it is necessary for groups of people to collectively review and edit data.

This chapter conducts two different case studies on the maturation phase of evolving tabular data's life cycle. Previous research has shown that crowdsourcing is a popular low cost and time-efficient method to collect tabular data for the initial collection phase [122, 173]. Without implementing quality control measures, crowdsourcing can yield inaccuracies with negative consequences [238]. One set of strategies we explore to verify and correct tabular data is to pay crowdworkers. At the same time, an alternative method includes continuous edits, or maintenance strategies, from unpaid contributors over 2.5 years. We conduct two separate case studies to examine how each method, paid crowdworkers, and unpaid contributors, improves crowd-collected tabular datasets' accuracy. The first case study focuses on what we call **verification strategies** and includes five strategies we adapted from literature [21, 97, 122, 151] that rely on posting micro-tasks to recruit paid crowdworkers to edit tabular data quickly. The second case study examines **maintenance strategies**, which include two strategies adapted from literature [220, 225, 231] that rely on waiting for unpaid contributors to continuously visit and make unsolicited and solicited edits to tabular data.

The two case studies in this chapter grew organically through our attempts to maintain a tabular dataset containing information on Computer Science faculty profiles since 2016 (chapters 3 and 4). While Wikipedia often features tabular data on individual pages, these tables contain a relatively small number of rows and columns. Our case studies feature larger tabular datasets consisting of thousands of rows and at least ten columns. Paid crowdworkers collected the initial dataset, and then we hosted it in a custom editable spreadsheet web application. To the best of our knowledge, no published work has studied the maintenance of a tabular dataset by unpaid contributions over multiple years. Instead, our second case study has attracted thousands of visitors who voluntarily corrected data and added new faculty information without monetary compensation. After observing this naturalistic maintenance by unpaid contributors, we explored how traditional methods using paid crowdworkers would work under similar circumstances. A verification phase using paid crowdworkers was started immediately after collecting a second dataset using paid crowdworkers to improve its accuracy. Equipped with two tabular datasets on the same topic, we analyze the strategies examined within each case study.

This work identifies the trade-offs of five individual verification strategies and two maintenance strategies for enabling groups of people to collaboratively verify and maintain evolving tabular data. We study the accuracy achieved and the payment or time required for human-driven "table augmentation" tasks (i.e., edits): fixing existing data, filling in empty cells, and adding new rows of data [239]. We also investigate the accuracy of different data types that may require varying levels of subjectivity and domain-specific knowledge.

An important metric for paid verification strategies is the monetary cost of an accurate edit. In contrast, an important metric for unpaid longitudinal maintenance strategies is the time needed to wait for an accurate edit and whether a few users make the majority of edits (similar to Wikipedia). We also examine if the number of visits, edits, or types of interactions can indicate whether unpaid contributors vandalize data during long-term deployment.

Overall, we find that verification strategies are more accurate at a lower cost when rehiring trusted crowdworkers. On the other hand, groups of unpaid contributors improved data regularly through small contributions with no signs of vandalism. They also excelled at editing subjective data types requiring domain-specific knowledge. The results of these separate case studies create discussions focusing on: (1) selecting appropriate paid or unpaid approaches to improve the accuracy of tabular data, (2) new hybrid approaches blending lessons learned from each case study, and (3) proposing new automated methods to attract unpaid contributions in peer production systems.

This chapter independently explores verification and maintenance strategies that enable collaborative editing behaviors. We avoid a direct experimental comparison between paid crowdworkers and unpaid contributors due to the inherent difficulty of acquiring naturalistic editing behavior and the constraints of running parallel longitudinal studies. Instead, by examining two naturalistic case studies on the same type of data, we gain insights into the behaviors and motivations of paid crowdworkers and unpaid contributors.

5.2 Overview of Case Studies

This work presents two case studies focusing on collaborative efforts to improve the accuracy of evolving tabular datasets, one of the most fundamental ways of organizing information [74]. This section explains the structure of the collected data and the study design considerations for each case study. Our Institutional Review Board classified the procedures as exempt from review. The requesters, paid crowdworkers, and unpaid contributors were informed and consented to their data being used for research.

In our setting, data verification strategies rely on paid crowdworkers recruited on Amazon Mechanical Turk to edit inaccurate data. Alternatively, data maintenance strategies rely on unpaid contributors who visit a publicly-accessible web application with an editable spreadsheet and make edits over time.

5.2.1 Spreadsheet Interfaces and Tabular Datasets

In the case study on maintenance strategies, unpaid contributors used a publicly-accessible web-based editable spreadsheet interface, Drafty. The version of Drafty used in this chapter is slightly updated from its original version in chapter 5. The most notable change is adding a new feature to add new rows to the data. In the case study on verification strategies, paid crowdworkers used Google Sheets. Following prior recommendations [57], all interfaces used built-in inputs to validate data. Each case study used a distinct instance of a tabular dataset of Computer Science faculty academic profiles from top programs in the United States and Canada. Each academic profile is a row in a spreadsheet. Each column corresponds to their affiliated university, the year they joined as faculty, rank, subfield area of expertise, where they received their Bachelors, Masters, and Ph.D. degrees, and sources used to gather the information. During both case studies, the profiles also featured gender at the request of researchers who wanted to analyze hiring trends.

5.2.2 Study Design Considerations

The case study on maintenance strategies spanned years and grew organically in a dataset we hosted that naturally attracted visitors who contributed to it without any payment. Therefore, we made specific design decisions to study verification strategies in a naturalistic setting. This section covers how the data was initially collected and our study design decisions to study realistic editing behaviors.

For each case study, undergraduate and graduate students from two separate human-computer interaction seminars acted as requesters, as part of a graded class assignment, to employ crowdworkers to collect a different faculty dataset following the "Classic Micro" data collection strategy [173]. In this strategy, a task translates to finding all the information on a specific faculty member. Twenty students followed these methods in 2015 to recruit paid crowdworkers to collect the maintenance strategies' initial dataset, relying on unpaid contributors featured in chapter 3. In 2018, twelve students recruited crowdworkers to collect a new dataset for the case study of verification strategies. These same twelve students recruited crowdworkers again to complete the verification strategies. We did not reuse the dataset collected in 2015 for the case study of verification strategies; over the three years that elapsed, the data continuously evolved due to promotions and new faculty hires, among other naturally occurring events. A newly collected dataset in 2018 would contain errors due to mistakes made by crowdworkers during the collection phase or inconsistencies in online sources.

We, the authors of this public paper [224], did not conduct the verification strategies ourselves to avoid biases due to our experience with crowdsourcing and promote a naturalistic approach similar to those observed in the maintenance strategies. Instead, similar to Papoutsaki et al. [173], students within a class acted as requesters employing paid crowdworkers within the verification strategies. All requesters were given the same budget and two weeks as a time constraint for each verification strategy. Although each requester was assigned the same five verification strategies, they were free to experiment with payment combinations regarding pay-per-task, bonuses, and communication techniques with crowdworkers.

5.3 Case Study I: Paid Verification Strategies

We conducted an exploratory case study on paid data verification strategies for evolving tabular datasets by observing requesters employing paid crowdworkers in a naturalistic scenario in chapter 4. In contrast to prior work on data verification [21,97,106], our work features a real-world scenario where requesters have to balance the natural constraints of time and money to achieve results without the aid of a novel system.

5.3.1 The Five Verification Strategies

We defined five verification strategies by integrating the findings from chapter 4 with popular micro-task verification strategies from literature to edit tabular datasets. The strategies are summarized below and in Figure 5.1.

1. *Find-Fix-Verify*: A popular strategy introduced by Bernstein et al. [21]. Three sets of unique crowdworkers perform each task: the first set identifies errors, the second set fixes errors, and a final set verifies the information's accuracy.

2. *Find-Fix*: The first component of *Find-Fix-Verify*. Two sets of unique crowdworkers perform each task: the first set identifies errors and the second set fixes errors.

3. *Find+Fix*: The second component of *Find-Fix-Verify*. A single crowdworker is required to find and fix inaccurate data.

4. *Majority Rule*: Similar to *Majority Decision* proposed by Hirth et al. [97], the most common response per data point was deemed correct. Unique sets of crowdworkers redundantly collect sets of data until two or more sets are in agreement.

5. *Expert Rule*: A variation of *Majority Rule*, was inspired by Marcus' [151] tactic of enlisting a trusted crowdworker as an "expert" to review tasks performed by others to ensure their accuracy. Unique sets of crowdworkers redundantly collect duplicate sets of data. A third crowdworker then compares the multiple sets to determine the correct data.

In this paradigm, trusted crowdworkers might not possess domain-specific knowledge but might have previously completed tasks for the requester and are thus deemed an expert. The ability to re-recruit trusted crowdworkers as experts follow Daniel et al.'s [57] recommendation for requesters to develop long-term relationships with crowdworkers. In our study, requesters employed their methods to recruit trusted crowdworkers on a task-by-task basis per strategy.

We limited the number of verification strategies to five following the advice from Wiggins et al. [230] who found a negative correlation between the number of verification techniques used and money paid after analyzing a collection of

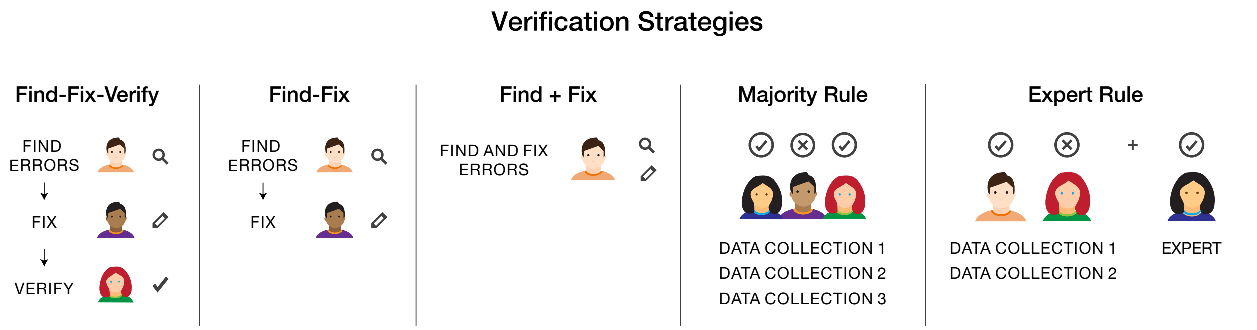


Figure 5.1: The five verification strategies outlined above require paid crowdworkers for each verification step. For example, *Find-Fix-Verify* and *Find-Fix* require a unique crowdworker for each step, whereas a single crowdworker performs *Find+Fix*. Both *Majority Rule* and *Expert Rule* rely on redundant data collection, while *Expert Rule* relies on a trusted crowdworker to review data and break ties.

citizen science experiments. Because requesters in our case study had to follow a strict budget for spending, we wanted to limit the number of strategies that could negatively affect the outcome.

5.3.2 Method: Verification Strategies

In Spring 2018, twelve students from a human-computer interaction seminar acted as requesters as part of a graded assignment. Each student (requester) employed paid crowdworkers to verify a crowd-collected tabular dataset over two weeks. Requesters used all five verification strategies and were randomly assigned one university per strategy. We removed three requesters' data from our analysis for not following the study procedure.

Requesters balancing crowdworker compensation, quality, and time

Requesters had to balance their budgets for verification tasks. Some of these tasks were repetitive, such as those in *Majority Rule* or *Expert Rule*, and required duplicate data for verification. Requesters naturally accounted for this strict 2-week constraint, echoed by Faridani et al.'s [70] recommendation, by balancing compensation against desired completion time. Therefore, it is difficult to forecast costs for tasks focusing on large datasets requiring repeatable tasks. For example, Mason and Watts [152] discovered that an increase in payment per task results in an increased quantity of work, but not necessarily quality. However, there is a balance between payment and quality to be discovered per task type, as Ho et al. [98] demonstrated higher payment increases quality. Hence, our requesters undertook a testing phase to learn what level of payment was appropriate to ensure quality results.

Initial steps and recruitment of crowdworkers

First, requesters read three seminal works on crowdsourcing [21, 68, 122] and watched a talk by Marcus on working with crowdworkers [151]. Requesters were randomly assigned five universities to collect and then verify data for using paid crowdworkers and a unique verification strategy per university. We checked the number of professors requesters would need to collect and verify, ensuring a balanced sample. Each requester received $50 in credit to use on Amazon Mechanical Turk, informed by the difficulty of tasks and recommendations from [173]. The experiment consists of three phases: testing, data collection, and data verification. Requesters were required to spend $5 in total for testing. During testing, they could experiment with Amazon Mechanical Turk and develop, test, and review each strategy's successes. Then, the requesters had $4 per university to spend on Amazon Mechanical Turk to collect data. After that, they had $5 to spend for data verification on Amazon Mechanical Turk as well. During the verification phase, requesters could experiment with payment structures but could not use preset qualifications.

How edits are made

Requesters hosted each dataset per university on a separate instance of Google Sheets to allow crowdworkers only to edit data for the assigned verification strategy. Verification was only performed after the collection phase was complete. We chose not to direct crowdworkers to the existing web platform used to study maintenance strategies to ensure that unpaid contributors and paid crowdworkers could not access each others' datasets. Each verification strategy could influence if a single worker edited multiple pieces of data (Majority Rule, Expert Rule) or reviewed data for possible errors (Find-Fix, Find+Fix, Find-Fix-Verify).

Verification Strategy	Total Labeled Edits	Total Edits	Total Payment	Payment per Edit	Payment per Accurate Edit	Payment Increase Accurate Edit
Find-Fix-Verify	190	298	$67.97	$0.23	$0.32	42%
Find-Fix	200	472	$73.19	$0.16	$0.27	71%
Find+Fix	197	311	$44.74	$0.14	$0.23	60%
Majority Rule	222	674	$44.77	$0.07	$0.11	61%
Expert Rule	207	529	$54.63	$0.10	$0.15	48%
All	1,016	2,284	$285.30	$0.13	$0.19	56%

Table 5.1: A summary of edits made by paid crowdworkers in the case study of verification strategies. It includes the number of edits we manually labeled as correct/incorrect out of all edits, the total amount of money spent on verification strategies, the payment for each edit, and the increase required to obtain an accurate edit. *Expert Rule* required the least amount of additional money to generate accurate edits.

5.3.3 Results: Verification Strategies

Requesters spent a total of $285.30 to verify data, generating 2,284 edits at the average cost of $0.13 per edit. The total costs include payment-per-task and bonuses. To compute each strategy's accuracy, we first used stratified sampling to select edits to label as correct or incorrect. We compared each edit's value when it was made with faculty web pages, LinkedIn profiles, and resumes. We labeled 1,016 edits by hand, shown in Table 5.1. The overall accuracy for verification strategies (64%) reported in Table 5.2 is computed by summing the total correct edits per verification strategy over the sum of the total edits. To understand the cost needed to generate a correct edit, we compute cost-per-correct-edit by dividing the cost-per-edit over the total number of correct edits, as reported in Table 5.1. The average cost per correct edit, \$0.19, is 56% higher than the cost to generate an edit. The higher the accuracy, the more money goes toward generating accurate data.

Trade-offs between Verification Strategies

Each strategy has advantages and disadvantages. *Majority Rule* and *Expert Rule* generated accurate data at less cost compared to strategies adapted from *Find-Fix-Verify* (Table 5.1). This shows that collecting duplicate data is a cost-effective method. *Find-Fix-Verify* has the highest overall accuracy across all strategies at 70%, and *Expert Rule* is second, at 69%. Both strategies recruit an additional crowdworker to verify or break ties. This extra step increases costs but generally leads to better accuracy overall and per column. This finding points to a future question on the number or quality of crowdworkers needed to review information until it is correct.

Accurate strategies allow requesters to generate correct data and control costs. For example, *Find-Fix*, the least accurate strategy, required a 71% increase in cost to generate correct edits. *Find-Fix-Verify* increased costs because it recruits an additional crowdworker but yields a 17% increase in accuracy. The best strategy to control and decrease costs is *Expert Rule*. It achieves this at less than half the cost-per-edit than *Find-Fix-Verify*. These findings support the idea that hiring additional paid crowdworkers to verify data will improve its accuracy while reducing overall costs for maturing a tabular dataset.

		—— *Less subjective data types (columns)* ⟶					
Verification Strategy	**Overall Accuracy**	**Sub-field**	**Join Year**	**Rank**	**Masters**	**Bachelors**	**PhD**
Find-Fix-Verify	53%	56%	75%	72%	70%	91%	70%
Find-Fix	36%	52%	80%	63%	68%	81%	60%
Find+Fix	38%	62%	81%	61%	67%	83%	62%
Majority Rule	47%	59%	66%	57%	66%	68%	61%
Expert Rule	65%	59%	79%	70%	66%	78%	69%
All	48%	58%	75%	64%	67%	80%	64%

Table 5.2: Overall accuracy per strategy per column in the case study of verification strategies. *Expert Rule* has the highest overall accuracy when accounting for edits across all columns. Subfield, the most subjective data type, has the lowest accuracy per data type. Identifying a professor's subfield requires domain-specific knowledge and can be difficult for crowdworkers to interpret correctly. Accuracy is the average of the total correct edits over the total edits across all strategies.

Differences across Data Types

Each data type/column within tabular data has different properties that requesters had to consider. For example, a professor's subfield may require crowdworkers to have domain-specific knowledge to understand the differences between research areas. *Expert Rule* has the highest accuracy for subfield at 65%, as shown in Table 5.2. *Expert Rule* and *Find-Fix-Verify* outperform their simpler variations. Notably, *Expert Rule* produced higher levels of accuracy across every data type compared to *Majority Rule*. This observation shows that trusted crowdworkers might possess the requisite expertise and effort needed to correct information that is more difficult to find and understand. Results show that data types requiring domain-specific knowledge can benefit from multiple collection efforts, explaining why *Expert Rule* and *Majority Rule* outperform other strategies. In contrast, identifying where a professor received their Ph.D. or their rank are subjectively easier tasks, with an accuracy of 80% and 75%, respectively. Rank has three possible values: Assistant, Associate, or Full, while a professor's Ph.D. is often easy to find. Results show that easy-to-collect fields benefit from individuals reviewing the data while collecting duplicates could suffer from potential noise. Overall, we identify that different strategies best suit different data types. Future requesters should select the best verification strategy based on the occurrence or importance of a tabular dataset's data types.

Filling in Empty Cells & Adding New Rows

Filling in empty cells or adding missing rows is essential to creating an accurate dataset. Collecting duplicates is a quick and cheap method to generate more data, but duplicates may introduce noise. As shown in Table 5.3, our data shows that *Expert Rule* is useful for filling in empty cells, possibly because an additional crowdworker can sort through this noise by selecting the correct value. We observed a drawback of relying on paid crowdworkers when adding new rows. They often incorrectly added rows for non-tenure-track positions (Lecturer, Adjunct, Staff, or Professor of Practice) or professors with appointments in a non-Computer Science department (e.g., Engineering, Media, Computational Biology).

Verification	**Overall**	*Filling in Empty Cells*			*Adding New Rows*		
Strategy	**Accuracy**	**N**	**Labeled**	**Accuracy**	**N**	**Labeled**	**Accuracy**
Find-Fix-Verify	53%	118	91	66%	90	57	77%
Find-Fix	52%	156	109	63%	20	13	54%
Find+Fix	62%	101	62	56%	44	35	80%
Majority Rule	47%	222	94	55%	98	65	68%
Expert Rule	65%	239	111	72%	24	19	63%
All	48%	836	467	63%	276	189	71%

Table 5.3: Overall accuracy across verification strategies, and filling in empty cells, and adding new rows. *Find+Fix* has the highest accuracy for newly-added rows, while *Expert Rule* was the most accurate for making edits to empty cells. Accuracy is the average of the total correct edits over the total edits across all strategies.

Paying for Correct Edits

Managing a strict budget to build an accurate tabular dataset can be difficult. Using strategies with high accuracy can ensure requesters do not need to post additional micro-tasks to acquire accurate data. Overall, in the verification strategies, requesters spent $0.13 per edit to recruit crowdworkers, as shown in Table 5.1. These numbers increase when accounting for accuracy: requesters paid crowdworkers $0.19 per correct edit when posting micro-tasks. This 56% increase in the money needed to acquire a correct edit can make it difficult to predict the exact cost to verify an accurate dataset. To keep costs predictable, we recommend using a strategy such as *Find-Fix-Verify* or *Expert Rule* that employs a trusted worker to perform a final verification step.

5.3.4 Takeaways: Verification Strategies

The closest dataset to ours [173] includes faculty profiles that were collected and not verified. In our work, we only focus on the accuracy of the edits and not on the entire dataset. Thus, the accuracy we report (64%) is not directly comparable with the overall accuracy (74%) of the entire dataset presented in [173].

Contrary to similar prior work [173], we report accuracy per data type. For the verification strategies, these accuracies are consistently low. A possible reason for this is that data verification tasks might be more difficult than data collection tasks for paid crowdworkers. Verification tasks might involve correcting data that is more difficult to find and interpret because the initial data collection efforts were unsuccessful. Thus, these complex verification tasks do not resemble the more straightforward tasks paid crowdworkers often excel at [31, 73].

Verification strategies requiring an extra crowdworker to perform a final verification step, such as *Find-Fix-Verify* and *Expert Rule*, are more accurate. They perform better when correcting data that require domain-specific knowledge, confirming past findings of a similar relationship when using paid crowdworkers [86, 202]. Therefore, if a tabular dataset contains a large amount of subjective data, we recommend that requesters use verification strategies requiring an additional trusted worker.

Across the five strategies, *Expert Rule* controls costs the best. Its cost-per-edit is less than half than the second-best strategy *Find-Fix-Verify*. *Expert Rule* has an initial low-cost redundant data collection step, followed by a beneficial final verification from a trusted crowdworker. This mirrors Bernstein et al. experience when developing their implementation of *Find-Fix-Verify* to edit Word documents [21].

Expert Rule proved to be the most effective verification strategy to find and correct empty cells. Crowdworkers specifically recruited to perform this final verification have often seen the dataset before, making it easier for them to navigate and make edits. Their payment is also often increased per task, helping to explain that in addition to being a repeat trusted crowdworker, the higher payment can garner higher quality work [98].

In our verification strategies, requesters first generate the names of all the professors per university at a given point in time. If this initial step produced an incorrect list, this would cause the requester to run additional tasks. A requester could reduce these additional tasks if they run a task to verify this initial list of rows. Therefore, we recommend that future requesters using these verification strategies integrate this pre-collection task to verify they have the correct rows per entity to reduce the risk of running additional unnecessary tasks. As a parallel to managing data in a Knowledge Graph like Wikidata, this recommendation ensures data curators verify they had the correct number of nodes before paying crowdworkers to add additional metadata per node.

5.4 Case Study II: Unpaid Maintenance Strategies

The second case study on continuous unpaid data maintenance strategies is a longitudinal observation of unpaid contributors visiting and editing evolving tabular dataset. This "in the wild" study features two perpetual data maintenance strategies where unpaid contributors make either "unsolicited" or "solicited" edits over two and a half years.

5.4.1 The Two Maintenance Strategies

This case study grew organically from our multi-year effort to host a tabular dataset on Computer Science faculty where the data stagnates and needs additional edits. Wikipedia has shown that people freely visiting unstructured data are capable of successful maintenance [181]. Building off this idea, we present two continuous maintenance strategies to improve tabular datasets' accuracy, as shown in Figure 5.2. In one strategy, unpaid contributors maintain data by making "unsolicited" edits over time as they visit the online tabular dataset. In the second maintenance strategy, unpaid contributors maintain data when the platform hosting the tabular dataset "solicits" them to edit data matching their interests. The system analyzes their prior interactions with the tabular dataset to derive these interests. A unique aspect of these maintenance strategies is that the dataset curator waits for visitors to edit data over time. Hence, instead of paying crowdworkers to make edits immediately, these maintenance strategies can run perpetually by relying on a stream of edits over time by unpaid contributors who visit the tabular dataset in the wild.

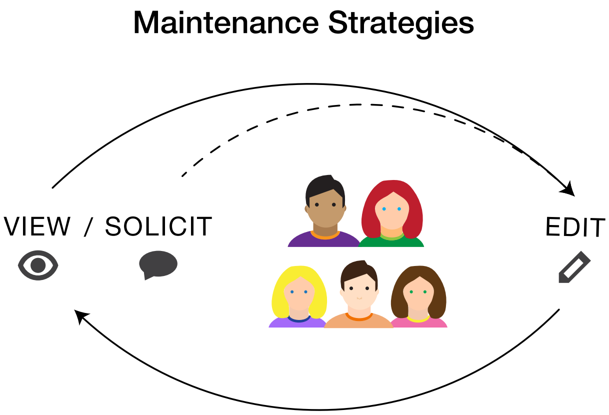


Figure 5.2: Maintenance strategies feature a continuous workflow allowing visitors to the tabular dataset to edit data repeatedly. Visitors, acting as unpaid contributors, can freely edit data of their choice or be solicited by the system to edit data that match their interests.

Relying on Unpaid Contributors

It is difficult for a curator to know when their dataset is out-of-date or inaccurate and immediately pay a crowdworker to correct it. Instead, our maintenance strategies rely on unpaid contributors visiting over time to review and edit data perpetually. We feel this is necessary because some tabular datasets, such as ours, are not static. The data can change and thus requires continuous review. Data could change because:

1. The initial information collected could be incorrect, and then it needs to be changed.
2. The initial information was correct, and then someone modified it incorrectly.
3. The data needs to be updated because it has changed. For example, a professor could change universities or change their subfield area of expertise.

5.4.2 Method: Maintenance Strategies

Starting in 2016, we conducted a case study on maintenance strategies in the wild using a publicly-available web application seeded with a crowd-collected dataset of 50,000 values from over 3,600 faculty profiles. We observed this human-centric approach of data maintenance for over two and a half years. We stopped running the case study when the web application hosting the dataset received a significant update in early 2019.

Initial steps

The maintenance strategies rely on waiting for unpaid contributors to edit and maintain the dataset. In our pilot studies on maintenance strategies for evolving tabular datasets, we found three recommended attributes of a tabular dataset that made it easier to attract visitors and study their contributions:

1. Each row of the dataset must remain valid for extended periods. For example, a professor can stay within academia for long periods. However, a faculty job posting could quickly become irrelevant once the position is filled.
2. Some columns within the tabular dataset should change over time. For example, a professor could change universities. This type of data presents more opportunities for edits.
3. The tabular dataset needs enough data to attract visitors to make contributions. If a dataset is missing too much information, users may not feel motivated to participate in something that feels neglected.

Recruitment of editors

Attracting interested visitors to make unpaid edits is necessary to study this continuous maintenance approach. When the system was initially seeded, we made posts across various CS forums and websites (Reddit CompSci, Hacker News, and TheGradCafe) to inform and attract an initial user base with related interests. ‘LabintheWild’ use similar strategies to attract users through social media platforms [186]. An example title used on posts was “Records of 3,600 computer science professors at 70 top universities (US/Canada) help us keep it up to date!" The goal was to appeal to users interested in Computer Science who may be:

1. a professor listed in the dataset,
2. a prospective student looking for an advisor,
3. a friend, colleague, or family member of a CS professor,
4. someone (e.g., in administration) who might be interested in running analysis on trends in Computer Science, or
5. someone who cares about adding to public information.

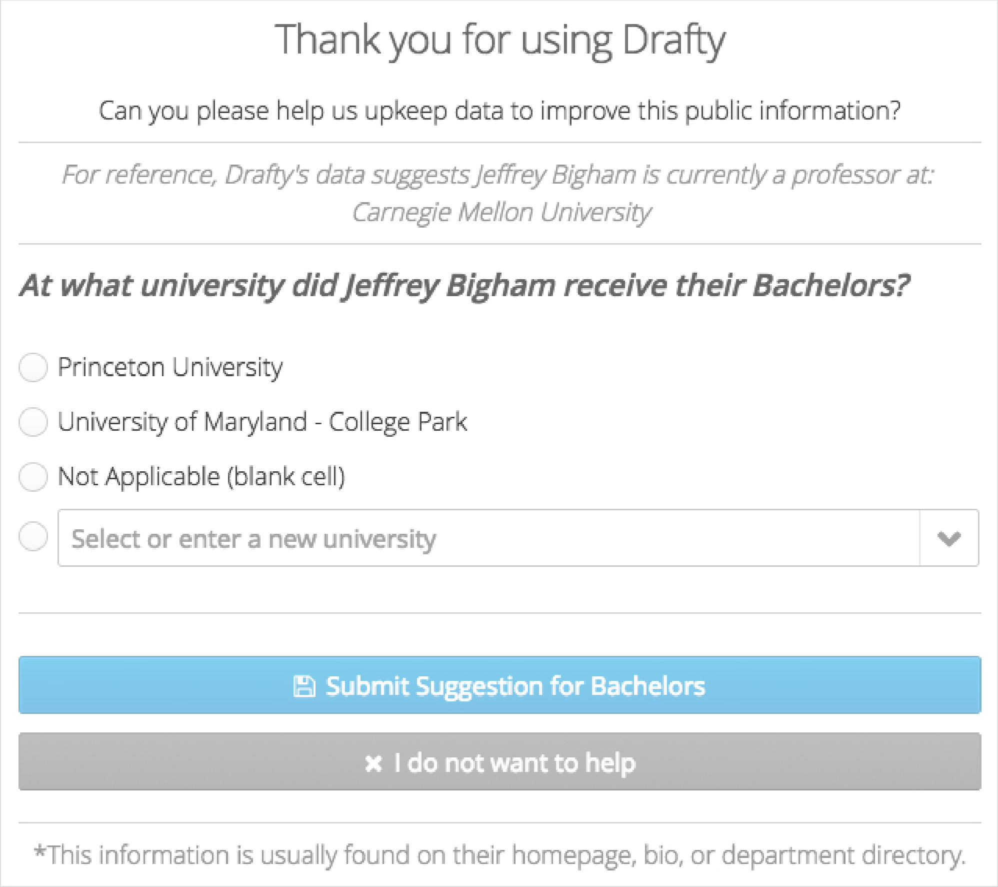
Each post advertising the dataset contained the text:

> Wanted to share a computer science resource a couple of us in the Brown University Human-Computer Interaction group have put together. It is a crowd-editable spreadsheet of data of approximately 3,600 computer science professors. For example, where they got their degrees, subfield of expertise, their join year and rank, etc... It might be useful if you're applying to Ph.D. programs or faculty positions, seeking external collaborators, or just to better understand hiring trends in CS departments.

We only made these initial posts. All subsequent traffic to the dataset was generated through organic search traffic or other means we did not control. While prior work shows that using social norms can motivate unpaid contributions, we chose not to employ this method [31]. Using such an approach may introduce additional factors affecting users' motivations within our naturalistic and longitudinal case study. Our goal was to assess the maintenance strategies and not to explore how best to build or attract a continuous flow of users.

How edits are made

The two maintenance strategies use an updated version of the public data system Drafty, described in detail in chapter 3. The strategies allow users to edit the data freely at any time without creating an account within the system. The system tracks users anonymously. However, the system presents users with a modal dialog informing them how to make edits on their first visit. Users can freely make *unsolicited* edits at any time by double-clicking a cell in the spreadsheet interface. Users can select a new value from previous edits to that cell, a predefined list of possible values, or freely enter text. The system does not require a special markup language for edits like Wikipedia. Specific data or features are not protected or semi-protected like in Wikipedia or Wikidata. The platform can also *solicit* edits from users by prompting them with a modal dialog request to fix a specific data value, as seen in Figure 5.3. The application solicits users to review a row of data matching their interests. The application infers interest by creating a "user interest profile" per user by tracking their interactions (i.e., search, sort, click, edit) within the spreadsheet and computing a



Thank you for using Drafty

Can you please help us upkeep data to improve this public information?

For reference, Drafty's data suggests Jeffrey Bigham is currently a professor at: Carnegie Mellon University

At what university did Jeffrey Bigham receive their Bachelors?

○ Princeton University

○ University of Maryland - College Park

○ Not Applicable (blank cell)

○ Select or enter a new university

*This information is usually found on their homepage, bio, or department directory.

Figure 5.3: This interface is the edit cell window from the Drafty web application that appears when Drafty solicits unpaid contributors to edit data based on their interests within the maintenance strategies. They can confirm the data does not exist, submit a correction, or exit to return to the spreadsheet interface.

relevance metric, like Wallace et al. [225]. The application displays the most recent edit as the correct value per cell in the interface. This is also how Google Sheets handles multiple possible values per cell in our verification strategies.

5.4.3 Results: Maintenance Strategies

The maintenance strategies were run over 1,025 days from May 26, 2016, to March 3, 2019. Visitors provided 2,651 edits at a rate of 2.6 edits per day. To compute the accuracy per edit, we labeled 1,020 edits by hand as correct or incorrect using stratified sampling following the same procedures with the case study on verification strategies. The findings are summarized in Table 5.4. We observed a common mistake where unsolicited visitors incorrectly edited a professor's Ph.D., as the university where their Bachelor's or Master's degree. We also observed the same error made by paid crowdworkers in the case study of verification strategies.

The overall accuracy for maintenance strategies is 89%, as seen in Table 5.5. Accuracy is the sum of the number of correct edits over the sum of the total edits across both maintenance strategies. The time between correct edits

Maintenance Strategy	Total Labeled Edits	Total Edits	Total Time	Time per Edit	Time per Accurate Edit	Time Increase per Accurate Edit
Unsolicited	958	2,566	1,025 days	9.6 hours	10.8 hours	13%
Solicited	62	85	1,025 days	12 days	13 days	5%
All	1,020	2,651	1,025 days	9.3 hours	10.4 hours	12%

Table 5.4: A summary of edits made by unpaid contributors (maintenance strategies), including the number of edits we manually labeled as correct/incorrect, and the total time spent waiting for edits. Soliciting unpaid contributors to edit data they are already interested in is the most efficient method to improve a dataset's accuracy, at a 5% increase in time to wait for an accurate edit.

		—— *Less subjective data types* ⟶					
Maintenance Strategy	**Overall Accuracy**	**Sub-field**	**Join Year**	**Rank**	**Masters**	**Bachelors**	**PhD**
Unsolicited	89%	91%	82%	90%	79%	73%	88%
Solicited	95%	100%	75%	67%	100%	80%	
All	89%	91%	82%	89%	82%	73%	88%

Table 5.5: Accuracy overall and per strategy per column. *Unsolicited* unpaid contributors excelled at correcting empty cells; this might be because of their prior knowledge of CS professors. Accuracy is the average of the total correct edits over the total edits across all strategies.

represents how frequently an unpaid contributor submits an accurate edit and is computed by dividing the number of days or hours per edit over the total number of correct edits (Table 5.4). In the maintenance strategies, the time between correct edits (10.4 hours) is 12% higher than the time needed to generate a single edit. This metric is comparable to a similar study by [225], where their users made 592 edits in a similarly structured dataset over 214 days at 75% accuracy. Their users generated a correct edit every 11.5 hours.

Our case study's extended length and frequent visitors could explain why our maintenance strategies have higher accuracy than studies with similar levels of edits per hour. Our case study's strategies produce minimal negative contributions (e.g., incorrect edits), supporting the idea that long-term maintenance is possible in public datasets.

Differences Across Data Types

Previous work shows users with expertise in specialized tasks or interest in the data perform better [202, 225]. Unpaid contributors may have an interest in or prior knowledge of the data, leading to higher accuracy when editing challenging information. This can help explain the varying levels of accuracy across the data types seen in Table 5.5 and suggests their domain-specific knowledge helps them accurately identify Rank and Subfield from personal web pages, publications, or their prior knowledge. In contrast, columns with less subjective data, such as Bachelors and Masters degrees, have lower accuracy. Upon review, these data types are more difficult to find: professors do not always list their degrees on their websites or other sources, whereas their Ph.D. is prominently displayed. Over our longitudinal study, 33% of participants edited at least one value for Rank or Subfield. While these data types require domain-specific knowledge, they can also change over time. This initial result points to the potential for unpaid contributors interested in the data to maintain these types of data over time. The maintenance strategies benefit from waiting for someone with domain-specific or pre-existing knowledge to assess those more complex data types.

Filling in Empty Cells & Adding New Rows

Table 5.6 shows that the accuracy for editing empty cells (93%) is higher than the accuracy for edits to cells with existing data. This observation might be due to unpaid contributors having pre-existing knowledge of particular rows in a dataset. For example, they can quickly correct an empty subfield because they already know a professor's area of expertise from reading their research papers. Prior knowledge can make adding a new professor to a university easy because they know that professor. Newly-added rows are 86% accurate, a level of accuracy similar to unsolicited users correcting existing data.

When users decide to contribute an edit, they could edit existing data, fill in empty cells, or add multiple new cells by adding a new row. Each of these requires varying levels of effort to complete. The following results count the number of times each user edits existing data, fills in empty cells, or the number of new cells they created when adding new rows of data. During our case study, 33% of unpaid contributors filled in mostly empty cells, while 42% edited cells with pre-existing data. The remaining 25% of unpaid contributors edited the same number of empty and non-empty cells. In a similar analysis, we compared the percentage of unpaid contributors filling in more empty cells than creating new cells by adding new rows. In this comparison, 42% of unpaid contributors filled in mostly empty cells, while 6% created new cells when adding new rows. The remaining 52% of unpaid contributors created the same number of new data points when filling in empty cells or adding new rows. This analysis shows unpaid contributors were actively filling in empty cells, but few primarily added new rows of data. Adding new rows of data requires more effort; thus, our user base often made edits requiring less effort. As a tabular dataset matures, its empty to non-empty cells balance will change. Thus, it is essential to view these findings within the context of a tabular dataset that is at the beginning of its maturity phase.

Maintenance	**Overall**	*Filling in Empty Cells*			*Adding New Rows*		
Strategy	**Accuracy**	**N**	**Labeled**	**Accuracy**	**N**	**Labeled**	**Accuracy**
Unsolicited	89%	1,219	524	93%	224	99	86%
Solicited	95%	35	20	87%	–*	–*	–*
All	89%	1,254	544	93%	224	99	86%

Table 5.6: Overall accuracy across maintenance strategies and for filling in empty cells, and adding new rows. *Unsolicited* unpaid contributors excelled at correcting empty cells. Their prior knowledge of CS professors possibly contributes to these high levels of accuracy. Accuracy is the average of the total correct edits over the total edits across all strategies. *Solicited users in the case study of maintenance strategies were not asked to add new rows.

Soliciting Users to Fix Data

Previous studies featured in Chapter 3 show that asking users to fix data they are interested in leads to more accurate edits [225]. Our web application solicited visitors 1,018 times to review and correct data matching their interests. They submitted 85 edits, with an accuracy of 95% (Table 5.5). In our study, soliciting visitors to review and fix data matching their interests leads to a 7% increase in accuracy compared to visitors making normal unsolicited edits. A two-tailed t-test of unequal variances shows this increase was statistically significant, $t(79) = -2.2$, $p = 0.03$. While soliciting visitors to fix data matching their interests proved effective, motivating visitors to complete these edits proved difficult. Only 7.5% of solicitations resulted in submitted edits. A related short-term study showed a similar rate of 8.8% of solicitations resulted in submitted edits [225]. These continued low submission rates across our long-term study show this is an area for future research. Later, we discuss future methods to increase these rates and provide more edits in less time.

Waiting for Correct Edits

This work shows the potential for continuous maintenance strategies to have consistently high accuracy levels, leading to predictable wait times to generate correct edits. While unsolicited, unpaid contributors submitted an edit every 9.3 hours, each correct edit was submitted every 10.4 hours. This 12% increase in the time to wait for a correct edit, as shown in Table 5.4, shows that consistently accurate edits can benefit continuous maintenance efforts. These findings show that if a dataset curator has the time to invest, maintenance strategies can generate an accurate dataset. Compared to prior work where users might be motivated by initial posts marketing a dataset or through gamified rewards [191], our results indicate a dataset's maturity phase can be extended by relying on users interested in the data.

Detecting Vandalism and Unpaid Contributor's Edits, Visits, & Interactions

This section reviews unpaid contributor's edits (frequency and content), number of visits, and total interactions (edits, clicks, searches, and sorts) to determine if vandalism occurred during our long-term deployment of the maintenance strategies. Previous research developed models to predict vandalism from user data such as edits and visits to open knowledge graphs in Wikidata [92, 220].

Our maintenance strategies use a custom web application to track interactions with tabular data in spreadsheets. In our study, 82% of unpaid contributors only submitted correct edits. While the accuracy of edits from only 15% of unpaid contributors was below the average accuracy of 89%, as seen in Table 5.5. An unpaid contributor's total number of edits did not correlate with their accuracy ($r = 0.15$). These results show that the maintenance strategies do not have to rely on power users (i.e., those making the bulk of edits) to create a mature accurate tabular dataset. This result runs contrary to results from Wikipedia, where power users primarily maintain its unstructured data [96, 132, 181].

We did not observe unpaid contributors deleting data or entering values that would be considered inappropriate or incorrect. Errors most likely resulted from incorrect interpretations of data, such as a professor's Rank.

Our multi-year case study allows us to observe the frequency of edits over time compared to short-term work [175, 225]. Excluding the initial months when we made social media posts advertising the dataset, the months with the highest edits were November and December. The months with the lowest number of edits were June and August. November and December align with graduate school and job application periods, while the summer months are not overly active except for faculty members officially starting new positions. This observation shows further evidence of how the maintenance strategies can rely on the interests of unpaid visitors to extend the maturity phase by providing a valuable data source for a community.

An unpaid contributor's total number of visits did not correlate with their accuracy ($r = -0.11$). Also, 97% of unpaid contributors had more than 1 visit. This finding shows having multiple visits and making accurate edits aligns with results from similar work [92, 225]. An unpaid contributor's total number of interactions did not correlate with their accuracy ($r = -0.13$). This result indicates that a spreadsheet interface, which has been essentially unchanged in decades, might provide a familiar and easy to use medium to view and edit tabular data. We found no evidence of vandalism in the maintenance strategies after analyzing unpaid contributors' edits, their number of interactions, and visits. A possible reason for this is the maintenance strategies, and our tabular dataset is of interest to the community, thus providing community-based motivation to improve its accuracy.

5.4.4 Takeaways: Maintenance Strategies

Maintenance strategies relying on unpaid contributors produce consistently accurate edits. Our study's longitudinal nature demonstrates the potential to maintain existing evolving tabular datasets over long periods compared to prior work [3, 225] relying on unpaid contributors over short periods.

Unpaid contributors provided accurate edits per data type (column), regardless of whether they required domain-specific knowledge (i.e., Rank and Subfield), and thus builds upon past research of paid crowdworkers [86, 202]. This finding shows the potential to rely on user's altruistic motivations and mutual interests instead of paying crowdworkers to correct tabular data inaccuracies.

Unpaid contributors were highly accurate and active at filling in empty cells. Their prior knowledge could construe them as "experts", helping to explain why their edits increased accuracy [145]. This observation repeated itself whether the system solicited an unpaid contributor to edit data or not.

From our long-term observations as dataset curators running maintenance strategies, the unsolicited, unpaid contributors did not contribute new rows (i.e., professors) at a high enough rate to account for the number of new professors hired over three years. Similar research, either did not allow users to add new rows or the study was short-lived and did not allow for the data to naturally evolve over time to study if users will contribute enough new rows [175, 225]. Systems that run long-term maintenance strategies should reduce the effort required to add new rows of data through interface design and limit the minimum number of required fields to add a new row.

Soliciting unpaid contributors to review and correct data relevant to their interests increased edits' overall accuracy. This finding confirms prior short-term research [225] and provides a long-term mechanism to increase a tabular dataset's longevity. A possible reason soliciting based on unpaid contributors' interests is beneficial is that they solicited while completing natural information-seeking goals, such as finding a possible advisor [99].

Contrary to prior work on Knowledge Graphs and Wikidata [92, 131], our maintenance strategies, focusing on collaborative efforts editing tabular data in spreadsheets, show that the number of visits per unpaid contributor was not predictive of vandalism or inaccurate edits. Integrating tools to automatically translate and import tabular data to Wikidata [166, 177, 214], our work could generalize from focusing strictly on a spreadsheet interface to serving as a potential method to maintain tabular data from Wikipedia and Wikidata.

5.5 Results: Comparing Paid Verification and Unpaid Maintenance Strategies

This work conducts two case studies to study workflows for improving data quality by focusing on paid versus unpaid strategies. A common conception of crowdsourcing is the single-step paid verification process. However, another consideration could be adopting a continuous data maintenance framework where the verification happens over time by unpaid users. This chapter's case studies show the potential for continuous data maintenance by unpaid contributors to have higher accuracy across various data types than paid verification (Tables 5.3 and 5.6). Previous research has shown similar levels of accuracy for paid and unpaid workers performing similar tasks [27]. The domain-specific expertise of visitors in maintenance could contribute to explaining these differences.

When comparing both case studies, unsolicited unpaid contributors in the Unpaid Maintenance Strategies submitted a correction every 9.3 hours, whereas a requester spent \$0.13 per correction in Paid Verification Strategies. These numbers increase when accounting for accuracy. In this study, unpaid contributors submit accurate corrections every 10.4 hours, whereas requesters spend \$0.19 per accurate correction posting microtasks on Amazon Mechanical Turk. The Unpaid Maintenance Strategies required 4.6 times fewer resources to generate a correct contribution than the Paid Verification Strategies. If someone has the time to invest, the Unpaid Maintenance Strategies can potentially maintain a more accurate dataset over time. However, Paid Verification Strategies can be substituted when time is a factor, and money is not. Chapter 8 provides a direct comparison between unpaid contributors and paid crowdworkers replicating the results discussed in this chapter with with more validity by controlling for confounding factors such as seasonality, duration of study, and system used.

Both workflows are not immune to incurring monetary costs or additional time. While the web application to run Unpaid Maintenance Strategies incurs minimal server costs, paid crowdsourcing approaches are susceptible to time constraints [125]. For example, requesters have to approve completed tasks, post new tasks, and communicate with paid crowdworkers. In contrast, after the dataset is hosted, the Unpaid Maintenance Strategies only require time to wait for corrections. This approach does not require active effort to produce corrections, like the paid microtasks from the Paid Verification Strategies.

5.6 Discussion

This chapter examines two case studies on verification and maintenance strategies for improving the accuracy of evolving tabular datasets. Verification strategies require money to pay crowdworkers. We control the time to wait for edits in verification strategies by limiting them to a short 2-week period. In contrast, maintenance strategies require time to wait for an unpaid contributor to visit the dataset and make edits. We control for money with these maintenance strategies, as hosting the web application to edit the dataset is the only cost. Our initial findings inform future work that can directly compare paid micro-task work strategies and those that rely on unpaid efforts. We will discuss several future ideas in the context of our presented findings and prior work.

5.6.1 Selecting the Best Approach: Paid or Unpaid

Choosing to use a purely paid crowdsourcing approach like our verification strategies or to adopt a peer production system like our maintenance strategies is difficult. Verification and maintenance strategies are not immune to incurring monetary costs or additional time for the dataset's curators. While running maintenance incurs minimal server costs, in verification, the time required to wait for a crowdworker to accept a paid micro-task can depend on the payment [125]. For example, requesters in verification have to approve completed tasks, post new ones, and respond to paid crowdworkers. This might not be ideal during a continuous data maintenance effort lasting years. Nevertheless, this time is significantly less than the time required to curate a dataset using maintenance strategies. Maintenance strategies require waiting for edits or other methods to attract visitors passively. A dataset curator must know how much their time is worth and choose the best approach for their needs. If a dataset does not contain evolving data, it might be faster to use paid crowdworkers to verify existing data rather than wait. A better approach might be to mix intrinsic and extrinsic motivators in a single system.

Prior work has shown how mixing extrinsic and intrinsic rewards can improve the outcomes of specific tasks [31,73]. Our results reflect this prior finding, where paid or unpaid users can generate better outcomes for specific tasks. For example, intrinsically motivated users were more effective at open-ended tasks. For example, this chapter's results point to the idea that unpaid contributors are more accurate in editing data that is more subjective or difficult to find, like Subfield or Join Year.

Another alternative approach to hosting a system to maintain a tabular dataset is to use one of the many tools to automatically translate and submit tabular data to Wikidata [166, 214]. Future work could assess if this unpaid approach leveraging popular platforms and tools, such as Wikidata, is more effective than maintaining a tabular dataset on a smaller, homegrown platform. Our maintenance strategies are one example of using a lesser-known platform to host a tabular dataset focused on one topic. Future research could assess whether a tabular dataset's maturity phase would benefit more from a popular platform than a smaller homegrown system. From our experience developing a system to maintain tabular data, its most important advantage is the ability to customize and adapt features to a specific dataset quickly.

Information can be subjective and difficult to interpret and label [49, 67]. This can be especially difficult in tabular datasets containing data types requiring expert knowledge with no accompanying information like Wikipedia to help users quickly understand context [225]. Requesters in verification strategies had to evaluate paid crowdworkers editing existing data to re-recruit them for knowledge-specific tasks. In contrast, unpaid contributors' prior knowledge and interest benefited them when editing these data types. The verification strategies could benefit from deploying fast, low-cost tasks to pre-filter possible workers. A more ambitious idea would be for crowdsourcing platforms, such as Amazon Mechanical Turk, to allow requesters to find crowdworkers interested in the content of the tabular dataset. This would be analogous to our maintenance strategies benefiting from the user's interest in the data. By appealing to paid crowdworkers' intrinsic motivations and interests, they might be more motivated to find and interpret complex data types. This design implication also does not require requesters to develop more complex gamification mechanisms to improve results [138].

5.6.2 Hybrid Models to Compare Paid Crowdworkers and Unpaid Contributors

Our findings highlight future studies' potential to directly compare the collective efforts of paid crowdworkers and unpaid contributors' in controlled settings.

One method to exert more control is introducing errors to a tabular dataset intentionally and simultaneously recruiting paid crowdworkers and unpaid contributors and comparing their interactions, accuracy, and time required to correct errors. This would enable several valuable observations. For example, crowdworkers might take fewer actions to submit an edit since they want to progress quickly. In contrast, unpaid contributors might explore the dataset and only edit the information if they notice an error. Future work could expand the user interest profile used in the maintenance strategies with new types of interactions from paid crowdworkers and unpaid contributors. For example, the system could extract user interest from gaze information [95]. Running eye-tracking studies at scale in-the-wild is a recent advancement [174] to enable this type of research. With each piece of information carefully organized in cells, tabular data is an exciting interface to extract gaze information. Building on this idea, could each group's interactions before editing help determine their accuracy automatically? Automatically rating edits' accuracy to tabular data using interactions is an ongoing research question [225].

Could paid crowdworkers be motivated through targeted non-paid requests while editing to contribute to the tabular dataset? This is plausible if they believe these contributions are for the greater good. For example, Rogstadius et al. [192] studied how intrinsic motivators, such as completing tasks that contribute to a knowledge base, positively influence work quality compared to extrinsically-motivated tasks, such as monetary incentives. On the contrary, could unpaid contributors be motivated through a pay-per-task model? Previous researchers introduced an extrinsic motivator as an intervention for initially intrinsically motivated users to contribute [116, 228]. They found that this intervention did not impact the quality of work. Thus, using money to extrinsically motivate users of a public system might not be beneficial for the maturity phase of the system's information.

While our verification strategies rely on money to extrinsically motivate paid crowdworkers, our maintenance strategies more so resemble "micro-volunteering." Bernstein et al. define micro-volunteering, an example of altruistic motivation, as the process of completing small online tasks for social good [18]. Our unpaid contributors possibly visit and contribute edits because of an interest in Computer Science. While we did not explicitly advertise the system to our friends, they possibly have visited the dataset. If we recruit our friends, this would resemble "friendsourcing," where volunteers are recruited from a network of friends to make voluntary contributions [29]. Another possible source of unpaid contributors is the professors themselves. We have received numerous requests for copies of the data so others can analyze hiring trends. Thus, it is reasonable to assume that some professors might be editing their information. We view this as a potential benefit of maintaining a dataset of public information that individuals might be personally interested in. Future work could compare unpaid contributions from different recruitment strategies, like micro-volunteering or friendsourcing. It would be reasonable to assume that friends share similar interests and might freely contribute to datasets that are relevant to their friends. However, it is unclear whether friendsourcing is adequate for maintaining a tabular dataset's accuracy in the long run.

Another approach to tease apart the motivations for unpaid and paid contributions is randomly asking users why they contributed after an edit. For example, maybe they are correcting their data, or they were paid to correct data. This in-the-moment feedback method has proven successful in prior research [35, 223] and could be helpful to understand user's intrinsic and extrinsic motivators. Also, these motivators might differ across tabular datasets covering different topics.

These future directions build on our findings and prior literature to create hybrid strategies to maintain accurate datasets. The ability to continuously and cost-effectively recruit the right editors could enable tabular datasets to be perpetually maintained.

5.6.3 Automated Methods to Attract Unpaid Contributors

One of our goals is to learn different approaches for creating self-sustaining evolving tabular datasets. The maintenance strategies do not need consistent effort to recruit and manage crowdworkers. However, they do require consistent numbers of interested visitors to contribute edits. Paying crowdworkers can generate edits quickly, but our work's initial observations show that this might be less effective than relying on unpaid contributors. We found it challenging to initiate a "start now" process for maintenance because the dataset's discovery is often serendipitous.

Wikipedia represents a success story in targeting and perpetually attracting unpaid contributions [181]. With free-form datasets such as articles, users can quickly and effortlessly discover facts from online sources of plain text, tables, or figures. In that process, they encounter incorrect or missing information. In contrast, users must perform multiple sorting and filtering operations with lengthy tabular datasets to find information. In this workflow, their focus rarely veers from their original query to new potentially interesting information, in need of edits, or missing information.

Large tabular datasets similar to those highlighted in this chapter that features hundreds to thousands of rows of data and multiple columns have a potential advantage compared to unstructured articles or small tables on Wikipedia. This advantage relies on analyzing a larger amount of data on a single cohesive topic, compared to analyzing smaller tables present on Wikipedia [126]. A platform hosting a single large tabular dataset can become more attractive to visitors by automatically generating and sharing insights and facts extracted from large amounts of data. This creation and sharing of knowledge could make a large tabular dataset more attractive, and potentially increase unpaid contributions [154].

Future work could create a system to generate sentences that describe insights and facts automatically based on statistical measures like maximums, modes, means, and outliers. This may also contain quantitative comparisons between the same values in columns or time-series data. These statistics could be automatically inputted into simple sentence templates, creating a fact to attract users. SuggestBot shows how to structure these fact-based sentences to maximize their effectiveness [53]. Structuring sentences in a "High-Involvement style" has also been shown to increase long-term retention among paid crowdworkers engaging in conversational micro-tasks [178]. By building on these ideas, one could structure these facts to attract, engage, and retain users. These human-consumable insights derived from large tabular datasets would move beyond prior research which focuses on finding tables related to other tables in Wikipedia [72].

A tabular dataset could market itself by automatically sharing these facts by posting them to relevant conversations in social media to attract regular visitors. In a similar effort, Botivist uses Twitter bots to recruit users to take action regarding Latin America's corruption issues [200]. This idea draws interesting parallels for recruiting unpaid contributors for maintenance. In our case, the system could share facts related to academia, Computer Science, or graduate schools to attract and motivate new users. If successful, this would automatically allow the maintenance strategies to attract unpaid contributors to perpetually tabular datasets.

5.7 Conclusion

This chapter shows initial evidence that paying people to edit data does not yield the same level of accuracy as relying on unpaid everyday contributors visiting a public data system. Someone's perception, for example, their level of interest in the data or task, potentially yields more accurate contributions. Chapter 8 provides additional evidence of a direct comparison between unpaid contributors and paid crowdworkers to build on the initial observations made in this chapter.

This chapter demonstrates how selecting the right verification strategy for different data types can yield value in money spent to acquire accurate edits. In particular, *Expert Rule* proved cost-effective for requesters, providing accurate edits at a lower cost than other paid verification strategies. It allows a dataset curator to improve the accuracy of a tabular dataset in a short period if the pay-per-task is high enough to attract and recruit trusted crowdworkers.

Continuous maintenance strategies enable groups of unpaid visitors to contribute edits freely or ask unpaid visitors to correct data that match their interests based on relevance. These approaches produced consistently accurate edits across all data types, indicating the potential of applying these approaches for self-sustaining tabular datasets. These continuous maintenance approaches can benefit tabular datasets where a row of data is viable for long periods, but individual cells may go out of date.

Accurate tabular datasets can yield exciting insights across fields, and ensuring their accuracy and longevity is essential to providing continuous insights to society. This chapter lays a foundation for different strategies to mature accurate, evolving tabular datasets. Questions remain about how to develop Drafty to serve its community of unpaid contributors better. Can additional methods and features be developed to motivate unpaid users to contribute accurate data?

Chapter 6

Sketchy: Motivating User Contributions during Short Sketching Tasks

This chapter presents Sketchy, a public data system where unpaid users simultaneously create, contribute, and automatically share their sketches during short 4-minute sketching tasks within virtual rooms. Users can see their peer's sketches evolve in real-time using the Peek feature. Users can freely initiate the Peek to select a random sketch they have not seen before. This behavior is intentional, and a lesson learned from the pilot studies. This system constraint was implemented so users could see all sketches within a room without overloading them with too many options. Given the short 4-minute sketching tasks, it was more conducive for the creative process to show one sketch updating in real-time compared to one screen with upwards of 100 randomly sorted static sketches. When using the Peek feature, users who see an inspirational sketch positively influence their next action to contribute new information back to the system. This chapter is a shortened version of [223], where I was the first author and responsible for all aspects of the user studies, system design and implementation, analysis, and the majority of the writing. As part of the contributions of this chapter, the Sketchy web application, the data gathered from almost 500 sketches, and the associated labels of whether they inspired a change in the peeker's sketch are publicly available[1].

6.1 Introduction

User experience design values sketching as part of the process, to act as a dialogue during ideation [32]. However, in a physical environment, the process of sketching and sharing typically do not co-occur, as space and materials constrain it. For example, simultaneous sketching commonly occurs in classrooms where students individually progress through

[1] Sketchy and the data are available at: https://sketchy.cs.brown.edu/

a given task where they rarely have the opportunity to collaborate and share their progress with their peers [9, 94]. Our preliminary observations show users physically peek (i.e., view another person's sketch) during in-person sketching tasks to gain inspiration. Instead of limiting students to only physically peeking at sketches from those sitting next to them, we enable digitally peeking at more random sketches within an entire classroom so people can simultaneously contribute new data in the form sketches while gaining inspiration from others' contributions in real-time.

To assess both how this digital "Peek" ability can affect creativity and which features of a "peeked" sketch are inspirational, we run two studies in a university-level user interface design classroom using a custom-developed web-based sketching application called *Sketchy*. In Sketchy, students sketch on a device they have at hand and can freely peek at their peers' in-progress sketches. To enable Peek in a digital environment, Sketchy automatically synchronizes everyone's sketches in real-time. This way, in-the-moment inspiration can come from a student sitting five rows behind, thus translating the physical experience of sharing sketches into a broader digital setting.

Classroom sketching reveals behavior where such peeking is part of the design process, as Loksa [141] reports, "we observed students peeking on the canvases of other group members, even while still working on their own task." But why? Past research has shown inspiration can "fire the soul", playing an essential role in the creative process [90, 168]. Determining how creative or inspired a student may be is a difficult task, given the inherent ambiguity and subjectivity of these concepts. In the following paragraphs, we define our notions of these concepts as they will be used in developing Sketchy and the Peek feature.

We define creativity as mini-c (personal level) creativity, as derived from Beghetto and Kauffman's taxonomy [113]. Mini-c creativity recognizes "that intrapersonal insights and interpretations, which often live only within the person who created them, are still considered creative acts." To evaluate "personal-creativity", Sketchy aligns itself with the Müller-Wienbergen et al. idea of an Individual Creativity Support Tool [161]. This focus on the individual differs from the group creativity support of 6-3-5 Brainwriting [189] and C-Sketch [204], where an individual only contributes to the group's final design. Similar to past studies of individual creativity support tools identified by Seidel et al. [203], our evaluation compares digitally peeking versus a baseline condition of physically peeking at another sketch using the Creativity Support Index (CSI) [37, 47], to measure a tool's ability to support creativity, similar to the NASA TLX [91]. Our usage of the CSI follows the recommendations of Wang et al. [229] and prior work by Benedetti et al. [14].

In our preliminary in-person sessions, we observe users peeking their peers' in-progress sketches to gain inspiration for their in-progress designs. Past research and our pilot studies show inspiration is too subjective to analyze quantitatively [213]. Asking a third party or a computational model to label inspiration in real-time is difficult to scale; they are equally unable to reveal what a sketcher is thinking. Why did they change something, and what part of a sketch inspired them? Since one cannot ask the sketcher if they were inspired in-the-moment, could a more direct question serve as a proxy?

Sketchy's idea of in-the-moment inspiration aligns with Cropley's [54] idea that inspiration is "hitting on a solution."

From this notion, we define "inspiration" as the process of being stimulated to *imitate* an idea from another sketch. Our definition of inspiration alludes to the idea of intended change. Imitation is difficult to precisely quantify and therefore requires the use of a simple proxy—in this case, posing a direct question to users while they were digitally peeking: "Will you change your sketch based on what you see in this sketch?" Subsequently, we quantify the actions users perform when going back to their canvases and relate them to their answers to this question.

Our direct question follows the creative process explained by Oleynick et al. [168], where the "process of being inspired by gives way to the process of being inspired to, which motivates action." Following peeking, a user's subsequent action represents the "actualization" of their inspiration to bring a creative idea to life, as defined by Thrash et al. [217], in ways such as peeking again, sketching, or erasing parts of their sketch.

Across the two studies within this chapter, we investigate the nature of creativity and inspiration based on a user's ability to digitally peek at the nearly unlimited number of sketches being drawn concurrently by their peers. We utilize both Sketchy and the Peek feature in classroom sketching tasks to answer several research questions:

R1) Does having the option to digitally peek lead to higher creativity measures and satisfaction with students' own sketches?

R2) When do students peek, what happens when they peek, and how often does peeking inspire them to change their sketch?

The answers to these questions explain the Peek feature's pedagogical contributions to the shared real-time classroom experience, how digitally peeking supports individual creativity, and the potential for a personalized version of Peek.

Our studies show a digital form of peeking compared to traditional physical peeking better supports creativity by increasing users' sense of collaboration, freedom of expression, and feeling of satisfaction that the reward is worth the effort of sketching. Peeking itself is sometimes reported to inspire changes in a user's sketch. However, inspiration seems to come more reliably from sketches closer to completion, more detailed and carefully created by more experienced sketchers.

Our findings further show that Sketchy supports an individual's creativity by translating the benefits of physical peeking into a digital setting, thus empowering an entire classroom to both create individual sketches and gather inspiration from others' sketches in real-time. This is a crucial finding, showing how Sketchy can be a successful public data system where users quickly collect, contribute, and view evolving data in real-time.

6.2 Related Work

Understanding a system's community of users is essential to understanding what motivates their engagement and contributions to a public data system. This section reviews related work on sketching applications to support design in computer education and creativity support tools. These are the core fields that inspire the design of Sketchy and the implementation of the Peek feature. They are unique to this chapter and relate to the system implementation and factors unique to Sketchy's community of users.

6.2.1 Creativity Support Tools

Creativity Support Tools provide digitally-mediated assistance to enhance the creativity of groups or individuals. In Painting with Bob, Benedetti et al. [14] created a digital painting tool to increase creative expression and evaluated results using the Creativity Support Index [37, 47]. Sketch-Sketch Revolution [71] generates tutorials from sketches and workflows of experts to help novice users. Other systems provide specific instructions for users to follow. For example, ShadowDraw [136] dynamically updates a shadow image underlying a user's strokes (similar to tracing a background image), and Iarussi et al. [107] presented a drawing tool that automatically extracts construction lines to help users draw more accurately. In contrast, Sketchy provides a more leisurely approach and allows users the agency to look for inspiration until they are satisfied. This is a passive ideation approach rather than an active one, which would have required proposing specific actions.

Other Creativity Support Tools focus on educating users, rather than providing assistance, through intelligent tutoring systems that provide personalized feedback. For example, SketchTivity [233] aims to improve creativity by allowing instructors to provide real-time feedback on sketches outside the classroom. Building on the idea of real-time feedback, Keshavabhotla et al. [115] developed a system that applies instructor insights and observed pedagogical practices to develop progressive exercises, which use sketch recognition to give real-time feedback. Other tools blur the line between human and AI feedback on sketches, such as the text-based facilitator used by Walsh and Wronksy [226] to increase engagement with a sketching activity.

Our work does not focus on real-time feedback of sketches, but rather on providing students with a shared real-time progression through sketching tasks. This allows students to find examples from their peers that support their creativity in short tasks, e.g. about 4 minutes. This alleviates the need for instructor interaction or a trained complex model to provide this feedback.

6.2.2 Providing Inspiration in Iterative Design

Inspiration, recently the subject of many studies, is a key contributor in the creation process [43, 109]. How to support inspiration in art is an open question [219]. Past researchers explored real-time feedback mechanisms [165] in an attempt to support creativity in lab settings. Wang et al. [227] developed IdeaExpander, a system that supports group creativity by showing pictorial stimuli based on dynamic conversational content. Sketchy, however, is evaluated in classrooms rather than lab settings. Extending from prior work, Sketchy examines the role of co-inspiration among peers, as opposed to its role solely on the individual.

Another approach to generate inspiration is for designers to search through collections of prior examples. Juxtapoze generates a database of clip-art using the shape of the intended object [15], while Co-3Deator hierarchically focuses on 3D sketching by directly swapping constituent components for work done by peer designers [176]. Goucher et al. text mined crowdworker's written design solutions to extract inspiration stimuli that were later presented to participants [81]. In a classroom, it would be time-consuming for crowdworkers to pre-generate all possible stimuli, thus the Peek feature automatically provides these stimuli by allowing students to see random sketches from others simultaneously working on the same task.

Researchers have also ran pilot studies and searched repositories to provide sketching examples at different intervals as stimuli for participants. While Kulkarni et al. showed that repeated exposure to sketches improves the creativity of generated ideas, these ideas were only shown at fixed intervals [130]. Siangliulue et al. showed that allowing users to see examples when they want to, or "on-demand", is more effective in generating novel ideas [205]. The Peek feature builds on Siangliulue's work by enabling the "on-demand" condition in a realistic classroom setting [205]. Peek relies on a group of users to simultaneously sketch and provide their work as possible inspirational stimuli to their peers. A teacher could use Sketchy for any sketching task without running pilot studies or curating online repositories for inspirational stimuli. This real-time experience of co-creating and inspiring allows for the analysis of not only the sketch itself, but also of the creator. Sketchy's Peek feature builds on their idea of the "on-demand" condition. Additionally, the Peek feature generates labels and findings related to 2D stroke data adding to their contributions to "personalized examples."

6.2.3 Collaboration & Creativity: Shared Progression Encourages Learning

Past research efforts have focused on the benefits of a collaborative environment for improving efficiency and quality in artwork but have failed to study supporting inspiration [198]. Lee [135] aimed to increase collaboration and create complex artifacts by investigating the benefits of asynchronous interactions within real-time collaborations. Exploring these ideas in a paid crowdsourced micro-task environment, Gingold et al. [79] looked at averaging crowdsourced drawings to produce high-quality output from low-quality input. Limpaecher et al. [139] created a stroke-correction method to improve strokes in real-time using a crowdsourced drawing database. Finally, Lasecki et al. [134] created a prototyping tool allowing designers to quickly iterate and gather feedback by automating and crowdsourcing elements of their interfaces.

Researchers have studied asynchronous collaborative analysis in non-sketching settings. Goyal et al. developed SAVANT, a collaborative sensemaking web-based tool, which enables implicit sharing and knowledge synthesis through a post-it note interface [83]. Aandolina et al. uses a similar post-it note whiteboard interface for paid crowdworkers in order to gather real-time creative input during early-stage design activities which would improve brainstorming and concept mapping [7]. In another example, Goyal et al. study implicit knowledge sharing in a collaborative analysis tool [82], showing that participants retain more information and perceive the tool as more useful compared to when sharing is disabled. Sketchy compares its digital collaborative feature Peek against a baseline of physical peeking, thus building upon these ideas of knowledge-sharing in the context of an individuated collaborative sketching approach. Ren et al. proposes fostering a collaborative culture and acknowledges the importance of the individual stakeholder group. Similarly, Sketchy focuses on empowering the individual while also enhancing collaboration [187].

Previous studies emphasized enabling multiple people to collaborate on a single project, with the idea that a group of users will create better designs than individual users [204, 244]. In contrast, our studies show how to support an individual's creativity and satisfaction with their final design by allowing them to see their peers' sketches.

6.3 Preliminary Observations: In-Person Sketching

We ran two preliminary in-person sketching sessions to inform the design of the Sketchy application and its features. The goal was to qualitatively observe how people with a range of abilities performed over several tasks when given a choice to inspect others' work.

Participants were recruited via convenience sampling with occupations ranging from designers and software engineers, to product managers and directors. Participants were from the same large company; hence they could know each other. The two sessions consisted of 6 and 3 participants, respectively (40% female), with an average age of 30 years. Participants sat around a table and used a pen and paper to perform the following tasks:

(0) Sketch a low-fidelity mock-up of a sun. (warm-up task)

(1) Sketch a button a user can click.

(2) Sketch a 5-pointed star.

(3) Create a typeface (i.e. font) and write the word "Sketchy" with it.

(4) Perform a low-fidelity redesign of the Sketchy landing page.

Each task lasted around 3 minutes to simulate the experience of creating and exploring ideas. The idea is for participants to share in their sketching experience in real-time under the same constraints. Participants were not asked to view or imitate each other's sketches.

6.3.1 Peeking for Ideas and Inspiration

We observed every participant in the first session looking at other participants' sketches when their progress stalled. Participants said they were trying to gain insight on how to sketch a challenging aspect of the task when asked to explain this behavior. For example, in Task 1, P2 looked at P4's sketch because they wanted to imitate how P4's design captured the intention that the 3-dimensional button "needed to be pressed." Participants often spent more time looking at sketches that were either more complete or were drawn by someone more experienced (e.g., professional designers). During their design process, participants examined sketches at different stages of completeness, indicating peeking is potentially beneficial. They would often pause and look at their sketch and then slowly look around the room at others' sketches. This highlights the importance of the shared experience; participants shared their progression and sought ideas from their peers who had progressed further.

Behaviors supporting the importance of prior experience and level of completeness occurred during Task 2 (designing a typeface) when P3 said, "I might have an advantage because my spouse designs typefaces." After that statement, each participant focused on P3's approach and results. At the core, participants sought to gain ideas from others that they could take back and use to improve their in-progress sketches. This observation led to the creation of the Peek feature within Sketchy, aiming to serve as the digital equivalent to this physical behavior and avoid the constraints of physical environments or the number of participants.

6.3.2 Inspiration Leads to Imitation

Findings from the second session further emphasize the importance of viewing others' sketches in real-time during sketching tasks. Rather than brainstorming abstract concepts, participants looked for ideas they could observe and bring back to their sketches.

Each participant would sketch a specific concept, such as the spacing between letters, and then try to implement another concept. If their process stopped, they would imitate a concept from another participant through peeking. For example, in the middle of Task 2, P1 paused and tapped their pen on the desk, apparently contemplating their next step. P2 then imitated a concept (thicker lines) from P2 to improve their sketch, see Figure 6.1. When prompted, P1 felt that when viewing other sketches, "the process is more important than the result."

After the task, we wanted to explore what participants were looking for when physically peeking. As P1 explained, "Seeing what effect works gives me inspiration. If I do not see anything, I have to imagine it. Imagining is hard because it is not concrete." P3 described their ideal learning scenario as one in which "a person would be drawing and presenting it at the same time." Such observations informed the design decision in Sketchy to update the Peeked sketch in real-time. Also, the second session found that participants look for unique features in a sketch to find inspiration. P1 stated, "trying to create something too realistic proved difficult." P1 demonstrated they could improve their typeface design

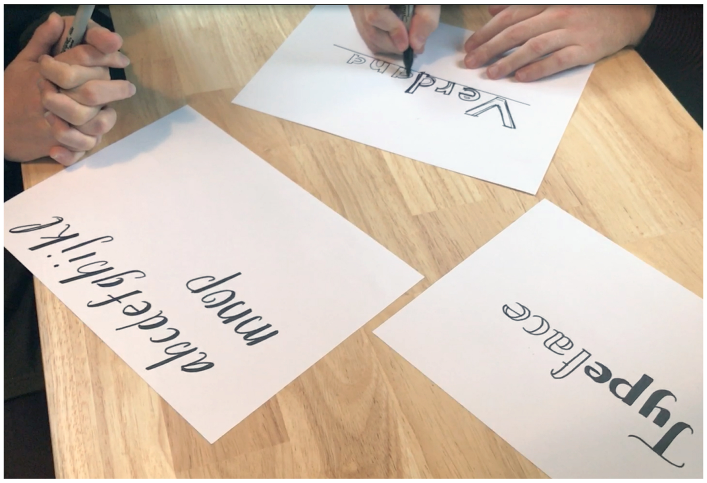


Figure 6.1: In session 2 of preliminary study, P1 improves design by imitating a feature they find inspiring from P2's typeface design. (P1: top; P2: bottom right; P3: left)

by imitating unique features that inspired them from P2 (Figure 6.1). From our observations, we aim for Sketchy to capture and share this *in-the-moment* inspiration through a digital form of peeking which enables a flexible co-learning experience where all users have equitable access to sketching examples generated in real-time.

6.4 Sketchy Public Data System

6.4.1 Design Considerations

In our preliminary observations, users seek inspiration from their peers to realize their creative ideas through a shared experience. This observation reinforces the idea that users can inspire each other as they progress through their designs.

The Peek feature takes a fundamentally different approach to support creativity: (i) Rather than having a group of students collaboratively create one final design, each student creates their own final design. (ii) Each student has control over their work and can freely choose to view others' in-progress sketches for inspiration. This user-first control allows students to work on similar tasks with their peers without relinquishing independent creative control.

We blended in-person observations with traditional design considerations from literature [14, 32, 205] to support student designers from a user interface design course based on the following principles.

Ease of Use: Designers sometimes struggle to begin a project. Sketchy features a straightforward user interface with a limited toolset, to serve as a simple starting point allowing for effortless design production [14, 32]. Moreover, Sketchy works across common platforms and allows users to draw strokes on a web-based canvas on both desktop and mobile platforms. Sketchy smooths strokes and renders the final output in Scalable Vector Graphics allowing for a consistent viewing experience across devices.

Freedom of Expression: User guidance and free-form flexibility are opposing goals. Within the confines of assistive technology, the user should feel as much freedom for creative expression as possible [14]. Unlike other studies such as ShadowDraw [136] and Sketch-Sketch Revolution [71], Sketchy does not dictate what the user should draw or what strokes they should make, allowing them creative freedom. While Sketchy provides a limited toolset, it does not impose structure. Instead, it focuses on empowering the user to sketch freeform and create and throw away ideas. Whiteboards and in-person sketching tasks support this user experience by providing erasers and multiple sheets of paper.

Supporting Creativity: The preliminary in-person study shows that participants can quickly become stuck on an idea without knowing how to implement it. Developing and witnessing many ideas in parallel can produce more diverse and higher quality results [66]. Conversely, choosing a design concept too early can result in design fixation [110], which can limit idea generation, even in experts [55]. The "Peek" feature can help reduce design fixation by exposing users to diverse ideas generated by their peers. Peek allows creators to passively share their in-progress work, like Mosaic [119], enabling users to see sketches evolve in real-time. The timing of seeing sketches in the creative process is also essential; seeing sketches earlier tends to be better for creativity [130]. Users are free to Peek at any moment to help kick-start their creative process since previous research has shown that allowing users to see sketches when they

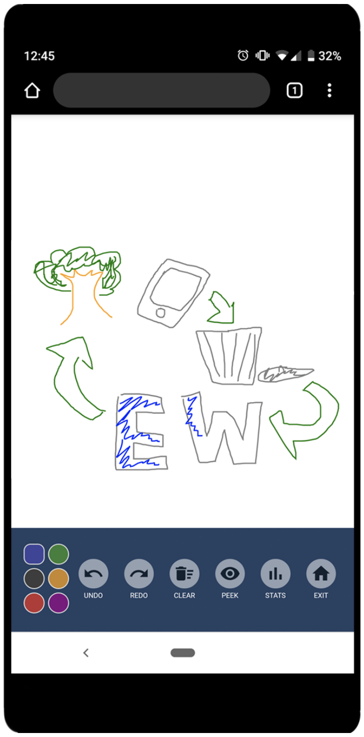


Figure 6.2: Users can draw, peek, undo, redo, clear, change stroke color, and view voting history of their sketch using Sketchy's interface.

want help to generate better ideas [205].

Building a Collaborative Space: In design education, designers typically share feedback in co-located studio settings [111]. Participants in design critiques often share their creative work with peers, exchange feedback and reflections, evaluate design approaches, interpret the concepts or artifacts, and brainstorm future possibilities [58]. Sketchy aims to scale the number of participants collaborating through the Peek feature by moving this experience into unlimited virtual space. This space streamlines the ideation and the exploration processes by allowing users to peek at their peers' sketches and then return to their drawings. Sketchy enables a consistent viewing experience across devices within this digital space by smoothing strokes and rendering the final output in Scalable Vector Graphics. This virtual collaborative "space" can enable the creation of classes with a dual format of remote and in-person students by allowing all students to sketch in a single virtual setting.

Individual results: Different users can create different results. Developing a personal style is critical to creative expression [14]. Sketchy, unlike C-sketch and skWiki, does not constrain users to drawing on another user's sketch [204, 244]. In contrast, users in Sketchy create their sketch without others directly changing their work. In Sketchy, users are free to explore their ideas and create different sketches, a key feature for creative expression (Figure 6.3).

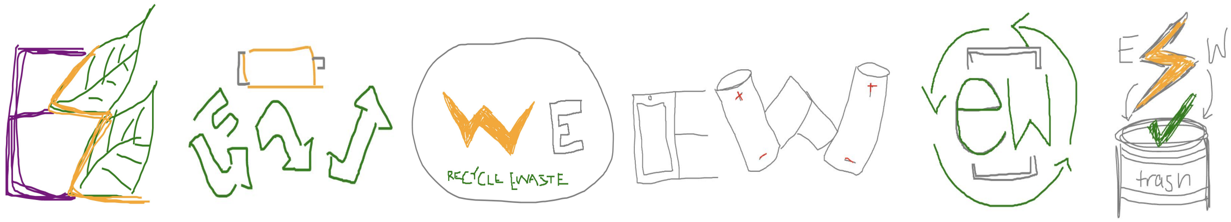


Figure 6.3: Study 2 - Task 2. Users "Design a logo for a company that disposes of E-Waste". Sketches are unique but can share similar concepts, from a lighting bolt and recycling symbols to batteries.

6.4.2 Implementation Overview

Based on the design considerations above, Sketchy offers a simple toolkit and functions on devices that users typically carry (mobile, tablet, and laptop), thus allowing for built-in real-time inspiration, which provides each user with their own sketching environment within an infinite virtual collaborative space. These features allow users to create low-fidelity designs through drawing on a web-based canvas (Figure 6.2) and to view sketches across different devices and screen sizes while retaining a sketch's intended resolution.

6.4.3 Sketching Input Methods

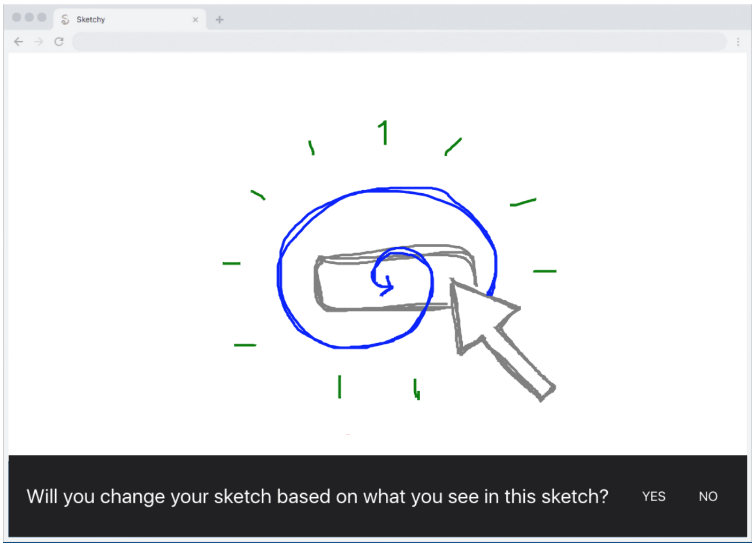
Sketchy has three different input methods. First, in **Drag Mode**, users click, hold, and drag with a mouse or touchpad to draw. In **Drag & Keyboard Mode**, users press and hold the **d**-key while Sketchy records all subsequent mouse movements as strokes. This mode lets users draw with greater precision since each hand performs one action, similar to how someone can draw using one-finger on a touchscreen device. Finally, in **Touch Mode**, users drag their finger to draw strokes, similar to finger painting.

6.4.4 Recording User Interactions

Sketchy records all user interactions: strokes, undo, redo, clear, peeking, and user votes, in a central database.

Sketchy's **draw** tool provides six colors for strokes: blue, green, gray, orange, red, and purple. The intention of implementing simple controls is to decrease the learning curve for the less experienced students in a class. Sketchy simulates a realistic sketching experience by rendering strokes resembling those made by a colored pencil. To simulate the imperfections of colored pencils, Sketchy first renders a fixed-width stroke and then creates smaller strokes with randomized opacity and offset from that initial stroke.

Sketchy has buttons for undo, redo, and clear functionality. Users can **undo** and **redo** all of their previous strokes. **Clear** allows a user to erase their canvas, an action which can indicate that a user wants to change their sketch entirely after peeking.



Will you change your sketch based on what you see in this sketch?

Figure 6.4: The interface users see when they Peek. The peeked sketch will update in real-time if new strokes are added or undone. To exit this interface they must answer the above question with either a "Yes" or "No".

6.4.5 Peek Feature

While working on the same task, the Peek feature enables users to view, or "peek", their peer's in-progress sketch. Users will see a peeked sketch update in real-time as their peer progresses, simulating a real-world sketching environment. It is common for users to peek at unfinished sketches. Peek selects a random sketch with at least one stroke and will not show the user the same sketch again unless they view all other sketches. This functionality mimics the observed behaviors from the preliminary in-person study. Due to system constraints, it would also be technically challenging to display upwards of 100 sketches evolving in real-time to a user.

To exit the Peek view (Figure 6.4), users choose Yes or No to the question: "Will you change your sketch based on what you see in this sketch?" The user's votes become labels corresponding to the in-progress sketches. From the earlier preliminary observations, we expect that when a student changes their sketch, this is a positive action, based on the intuition that one would want to change their sketch to make it better.

In sum, the Peek functionality enables the real-time generation and viewing of sketching examples. Unlike prior works, Peek allows users to simultaneously develop ideas and co-inspire with each other through the shared experience of using the same system under the same time constraints. Furthermore, in a classroom setting, peeking allows students to understand their peers' design decisions, sometimes even watching others change their minds and restart a sketch.

6.5 Method

We conducted two studies to answer our research questions. Study 1 addresses R1 with a between-subjects study design, while both studies evaluate R2. Study 2 addresses the descriptive question R3 through its larger sample size. Both studies demonstrate how Sketchy and its digital Peek feature can be used in different styles of classrooms. Study 1 is run in an informal lab setting with multiple round tables, while Study 2 is run in a large classroom (an auditorium in this case). By running these two studies, we can observe the Peek feature in the context of R2 across a diverse set of six sketching tasks versus three.

Students voluntarily participated in both studies (without earning course credit or a grade). They consented to participate in the study and share their data anonymously. Six $9.50 gift cards to a local restaurant were raffled off. Prior pilot studies informed the prompt for each task. Both studies lasted about 20 minutes, and besides the 1-minute warm-up tasks, students were given 4 minutes to complete each task and were never asked to peek. The data from the warm-up tasks were not used for analysis.

When Peek was enabled, students saw a slide showing how to use it. In our pilot testing, we observed students often physically peeking at the screens of others sitting next to them. Hence, when Peek was not enabled, we told students it was okay to look at their neighbor's screen. In Study 1, we could compare the results of physical versus digital peeking because all students sketch with the same digital drawing tools available in Sketchy. Our local Institutional Review Board approved these procedures.

Figure 6.5: Students sketching in a lounge during Study 1.

6.5.1 Study 1: Creativity Differences with the Peek Feature

Study 1 was conducted during a snack break (as depicted in Figure 6.5) between two different lab sessions of a university user interface course. It compared creativity measures with and without the Peek feature, using the Creativity Support Index [37, 47], between two groups of undergraduate and graduate students from a user interface class.

Students voluntarily signed up for the lab session of their choice. They had no prior knowledge of Sketchy. The first group of 58 students used a version of Sketchy without the Peek feature. The second group of 41 different students used Sketchy with Peek enabled. A digital form of peeking should theoretically expose students to more sketches than physically peeking at the screens of students next to them. We removed students from the analysis who did not complete all tasks, including the survey at the end. The analysis consists of 31 and 28 students, respectively.

The students were asked to perform one warm-up task to familiarize themselves with the system and then three design tasks:

(0) Draw a tree. (warm-up task)

(1) Draw an icon that represents saving to disk on a computer. No floppy disk icons are allowed.

(2) Draw a logo for a water utility company called Clean Water. The logo must contain the letters “C” and “W”.

(3) Draw a gesture that desktop computer users can trace out with their mouse in order to simulate a click, without having to physically click. The main challenge is for the design to be difficult to perform accidentally while being easy to perform intentionally.

6.5.2 Study 2: Peek Behavior and Inspiration in a Classroom

We conducted Study 2 at the end of a user interface design class with 115 students who volunteered to stay. From that initial group, 90 students completed all parts of the study, including a pre and post-survey. All students in this study used Sketchy with the ability to Peek at any time. This session tested the scalability of the platform and gathered more data from a larger sample size.

Some participants may overlap with Study 1, but the exact number is unknown because the data is anonymous. In this study, students performed four slightly different design tasks, for a change of pace:

(0) Draw an expressive face. (warm-up task)

(1) Draw an icon to represent drinkable water.

(2) Design a logo for a company that disposes of E-Waste. An added constraint was that the logo must have contained the letters “E” and “W”.

(3) Draw a pattern for a mouse to trace that will erase whatever is underneath the mouse cursor.

6.5.3 Study Procedure

First, students consent to the study procedures and answer the following multiple-choice questions using Likert scale responses:

(1) What is your level of sketch experience?

(2) What is your level of design experience?

(3) How likely are you to sketch ideas before starting a project?

(4) How effective do you feel viewing other Sketches would be for improving your own sketch?

(5) Which of the following methods do you typically use to gain inspiration for graphic design? The options for this question were “Looking at photographs of real-world objects”, “View images of designs on the Internet”, “Find a collaborator”, “Other”, and “None”.

During the study, when students digitally peek, they answer a question asking whether the sketch they saw would cause them to alter their sketch. During our pilot studies, we discussed different ways to determine if a user gained value from peeking at a sketch. Simply asking if the sketch provided value or inspiration was ill-defined and not specific enough for participants. Automatic estimates of whether an action came as a result of peeking are also challenging to define computationally. [213, 234]. Eventually, we decided to ask the user whether they intended to change their sketch after they peeked. This operationalization of *inspiration* seemed more manageable for users to judge quickly and served as a meaningful proxy for *inspiration* as an outcome. So after viewing a sketch, the user is prompted, “Will you change your sketch based on what you see in this sketch?”

We ran a pilot study to assess if participants should answer this question using a binary response (Yes/No) versus a 5-point Likert scale. We found no differences comparing each question type. Hence, we decided to use a simpler binary question for participants to answer when peeking.

After completing each sketching task, students rated how satisfied they were with their final sketch using a 5-point Likert scale. Finally, after completing all tasks, in a post-survey, students completed the Likert scale questions, assessed Sketchy by ranking the components for the Creativity Support Index [37,47], and answered the following questions:

(1) *Students in Baseline (No Digital Peeking):* How effective was viewing the sketches of people sitting next to you for inspiration?

(2) *Students who only used Peek:* How effective was the Peek feature for finding inspirational sketches?

(3) All students: How effective was viewing a sketch conceptually different from your own for inspiration?

(4) All students: How effective was viewing a sketch conceptually similar to your own for inspiration?

6.6 Results

In this section, we first evaluate the results of Study 1 to answer R1. Then, we present results from the small and large classroom studies, Studies 1 and 2, respectively, to answer RQs 2 and 3. Finally, to ensure a balanced sample, we evaluate the 2D stroke data from Study 2 to answer R3 since Study 2 had four times as many peeks as Study 1.

To investigate these research questions, t-tests were used to compare both the results between conditions, including Likert-scale responses, and the 2D stroke data following the guidance provided by past findings [11,167,212]. Unless otherwise specified, data passed both the Shapiro-Wilk test of normality and Levene's test of homogeneity of variances.

Our analysis is conducted per peek because different users are exposed to different stimuli at different times. Students see a new sketch at every peek because sketches are continuously updated. Each action (sketch, undo, redo, or clear) can cause a sketch to change while being peeked. Typically, we would need to aggregate multiple peeks on a sketch if multiple users saw the same version of that sketch simultaneously. We conducted an analysis that shows this did not occur.

In each task, sketches potentially change every second, exposing users to different stimuli. We would need to aggregate multiple peeks on a sketch if multiple users saw the same version of that sketch simultaneously. Our follow-up analysis shows that this did not happen.

The analysis is conducted per peek for several reasons. Notably, multiple versions of sketches were peeked at random times. This is because sketches are developed in real-time. Also, as we will discuss, multiple factors influence each peek. By analyzing per peek, we can include each factor to answer the research questions and inform a discussion for a future personalized sketch recommendation model.

6.6.1 Data Overview

Overall, across the two studies, 149 students completed all requirements and performed 447 tasks, sketching over 10,000 strokes and peeking 3,631 times. Across all studies, 61% of students used a mobile device (i.e., a phone or tablet). The next two sub-sections provide answers to possible questions about participation and device usage in both studies.

Having Peek Enabled Increased Conversation Rate

Students' data was only used if they completed the entire study. Participation was voluntary, and they could leave at any time. In Study 1, 53% of students who had Peek disabled completed all tasks, compared to 68% of students who had Peek enabled. When Peek was enabled, there was a 28% increase in the chance students completed all tasks. This result shows that Peek provides an increased sense of collaboration and can increase engagement across the same set of tasks. In Study 2 (large classroom) all students had Peek enabled, and 78% fully participated.

Controlling for Mobile Devices

The device used (mobile versus non-mobile) did not affect the timing or frequency of peeking or the number of sketches that inspired a design change per student in either study. Notably, peeking at a sketch made on the same type of device did not affect if a sketch inspired a design change. Hence, students can peek at sketches made on any device. These results show users can bring their own device when using Sketchy, and the distribution of mobile versus non-mobile devices will not affect the perceived quality of sketches.

	Study 1 (N=28)			**Study 2 (N=90)**			**Total**
Task	*1*	*2*	*3*	*1*	*2*	*3*	
Average Peeks per student	6	10	10	10	8	14	31
Total peeks	166	273	289	923	734	1,246	3,631
Peeks that led to an intention to change sketch	20%	22%	13%	16%	21%	14%	17%
Students who intended at least one sketch change	46%	75%	61%	36%	43%	46%	72%

Table 6.1: Summary statistics for Peeking in Study 1 (Peek Group) and Study 2 (Large Class). The **Total** column summarizes both studies. Task 0 was a warm-up and therefore is not analyzed. Peeking was frequent across all tasks, and students intended to change their design after peeking 18% of the time in Study 1 and 17% of the time in Study 2. These results for Study 1 only include the subgroup of students who used Peek. Students peeked more on average per task when there was a lower chance a peek would lead to a design change, indicating students kept searching for an inspirational sketch.

6.6.2 Digitally Peeking Leads to Creativity

Study 1 was a between-groups study that compared a group of 28 students with the Peek feature enabled to a group of 31 students without it. The students who did not have Peek enabled relied on physically peeking in the social setting of the lounge. Students reported their perceptions from peeking and completed the Creativity Support Index survey in an end-to-study survey. Creativity measures were higher in the group with Peek enabled, as seen in Figure 6.6, with the overall CSI score and three individual component differences being statistically significant.

A higher CSI score indicates better creativity support [47]; in Study 1, the overall CSI score is 72.3 for students with Peek enabled, compared to 64.4 for students without it. A one-tailed t-test reveals this 12% increase in CSI scores while using the Peek feature is statistically significant $t(57) = 2.19$, $p = 0.02$, $d = 0.57$. This result shows that the Peek feature adds creative value to Sketchy, especially compared to similar tools such as AutoDesk SketchBook Express, which received a CSI score of 64 [47].

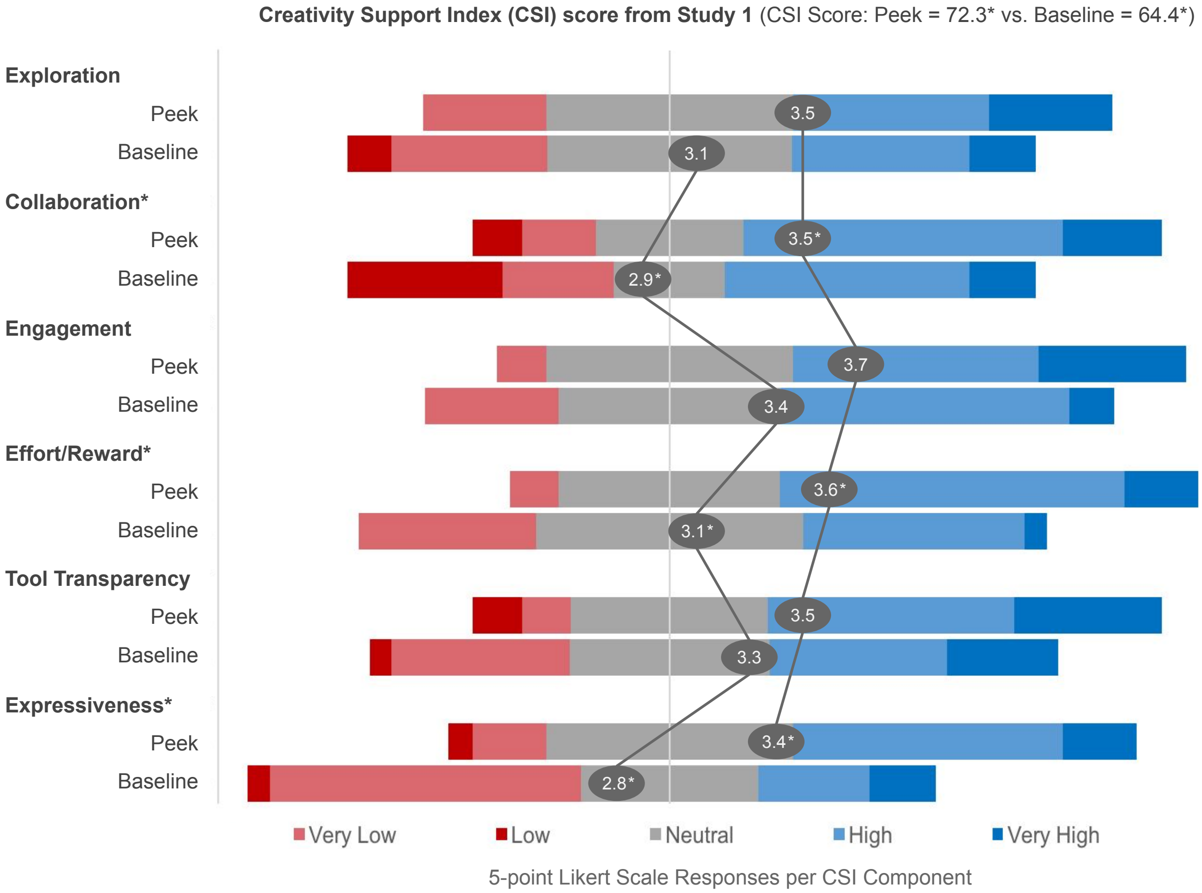


Figure 6.6: The CSI score from students who had Peek enabled in Study 1 was 12% higher than the baseline ($p = 0.02$). Each of the average CSI component scores and their distributions was higher than the baseline. Circles contain the mean per component – an asterisk (*) indicates a significant difference between Peek and the baseline. Colors per bar represent the percentage of participants who selected the Likert scale option. These results show that adding the ability to digitally Peek increased creativity.

Comparing the Creativity Support Index Components

The Creativity Support Index consists of six components, each rated on a 5-point Likert scale. The following three components saw a significant increase when Peek was enabled: Collaboration, Effort/Reward Tradeoff, and

Expressiveness.

Collaboration: "I was able to work together with others easily while doing this task." Students with Peek enabled rated the collaboration component 3.52 on average, while students without Peek rated it 2.94. A one-tailed t-test shows this 19% increase in perceived collaboration is statistically significant $t(57) = 1.73$, $p = 0.04$, $d = 0.46$. This result indicates peeking effectively facilitates collaboration and shows Sketchy supports the idea by Ren et al. to create a collaborative culture by adding functionality to empower the individual stakeholder group [187].

Effort/Reward Tradeoff: "What I was able to produce was worth the effort required to produce it." Students with Peek enabled rated the effort/reward tradeoff component 3.71 on average, while students without Peek rated it 3.13. A one-tailed t-test shows this 16% increase in perceived effort/reward is statistically significant $t(57) = 2.42$, $p = 0.01$, $d = 0.63$, indicating that Peek reduces the effort required to create a sketch. Since much of the effort involved in creating a sketch is coming up with an idea, this result indicates that Sketchy can effectively help users find better ways to integrate abstract concepts, see Figure 6.7.

Expressiveness: "I was able to be very expressive and creative while doing the task." Students with Peek enabled rated the expressiveness component 3.43 on average, while students without Peek rated it 2.84. A one-tailed t-test shows this 21% increase in perceived effort/reward is statistically significant $t(57) = 2.22$, $p = 0.02$, $d = 0.58$, indicating that Peek, by freely exposing students to a wider variety of ideas, allows them to find and express their own concepts better than physically peeking. This result supports the idea that the user should feel as much freedom for creative expression as possible [14].

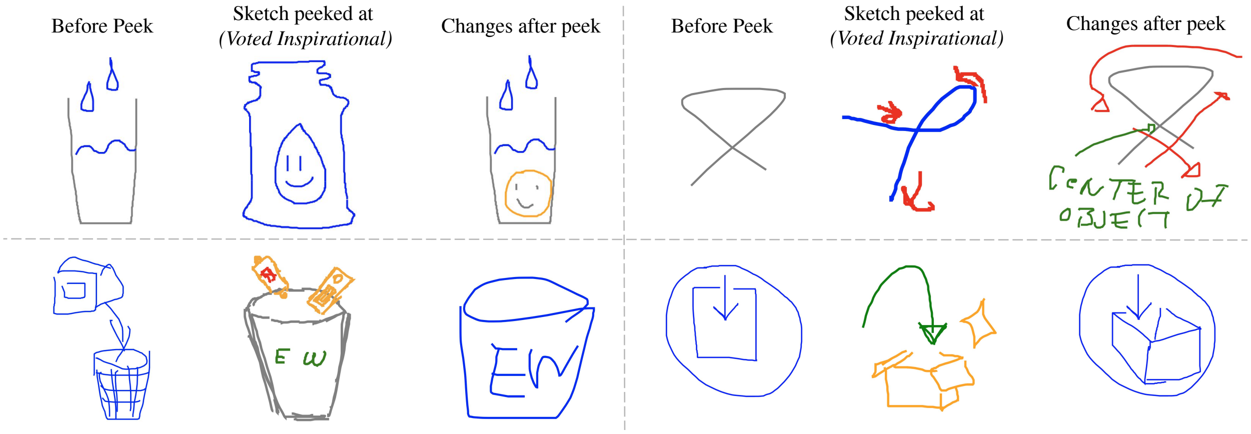


Figure 6.7: Examples of students imitating a concept from a peeked sketch they voted as inspirational. The top left sketch (Study 2, Task 1) imitates the smiley droplet. The top right sketch (Study 1, Task 3) imitates using red lines with arrows to indicate the cursor direction. The bottom left sketch (Study 1, Task 2) imitates the placement and inclusion of "EW." Lastly, the bottom right sketch (Study 1, Task 1) imitates the box's 3D dimensional shape.

Physical vs. Digital Peeking

Sketchy's Peek feature seeks to digitally mimic the act of physically peeking over one's shoulder. Students who had Peek enabled were asked "How effective was the Peek feature for finding inspirational sketches?" whereas students who did not use Peek were asked "How effective was viewing the sketches of people sitting next to you for inspiration?" Students from Study 1 found digitally peeking to be 40% more effective than physically looking at their neighbor's sketch for finding inspiration. A two-tailed t-test shows this increase was statistically significant, $t(57) = -4.3$, $p < 0.01$, $d = 1.13$. This shows that digitally peeking within a classroom setting can expose students to more ideas in a shorter amount of time than relying on finding ideas from the people sitting next to them.

After each task, students using the Peek feature were asked two separate 5-point Likert scale questions about their experience seeing conceptually similar and different sketches. Generally, students wanted to see sketches that were conceptually different from their own. In the post-survey, students who had Peek enabled were asked: "How effective was viewing a sketch conceptually different from your own for inspiration?" and "How effective was viewing a sketch conceptually similar to your own for inspiration?" Students from Study 1 believed that viewing a sketch conceptually different from their own was around 15% more inspirational. A paired two-tailed t-test shows this difference was statistically significant $t(27) = 2.01$, $p = 0.03$, $d = 0.49$. Also, students from Study 2 believed that viewing a sketch conceptually different from their own was around 11% more inspirational. A paired two-tailed t-test shows this difference was statistically significant $t(89) = 3.42$, $p < 0.01$, $d = 0.38$. These results hold across both small and large settings and across different tasks. A student might want to see a sketch conceptually similar or different during different stages of the creative process. For example, maybe it is more important to view conceptually different sketches during the ideation stage but conceptually similar sketches during the final stage in order to compare various implementations of ideas.

Students answered how satisfied they were with their final sketch on a 5-point Likert scale after each task. In Study 1, students who had Peek enabled rated their satisfaction an average of 3.60, while students without Peek rated their satisfaction an average of 3.11. A two-tailed t-test shows this 16% increase in the level of satisfaction is statistically significant $t(177) = 2.65$, $p < 0.01$, $d = 0.46$, indicating the inclusion of Peek led to students feeling more satisfied with their final design for each task. This result shows Peek can increase individual satisfaction by supporting the individual stakeholder, similar to Benedetti et al., instead of supporting the group's final design, such as C-Sketch and skWiki [14, 204, 244].

6.6.3 Peeking Occurred Often and Inspired Contributions

The large classroom study (Study 2) offered additional observations of what happens with peeking behavior in Sketchy at scale. This data, combined with Study 1, yields insights across a wider variety of sketching tasks. These quantitative insights focus on how often students peeked, when they peeked, what happened when they peeked, and how often peeking inspired them to alter their own sketch.

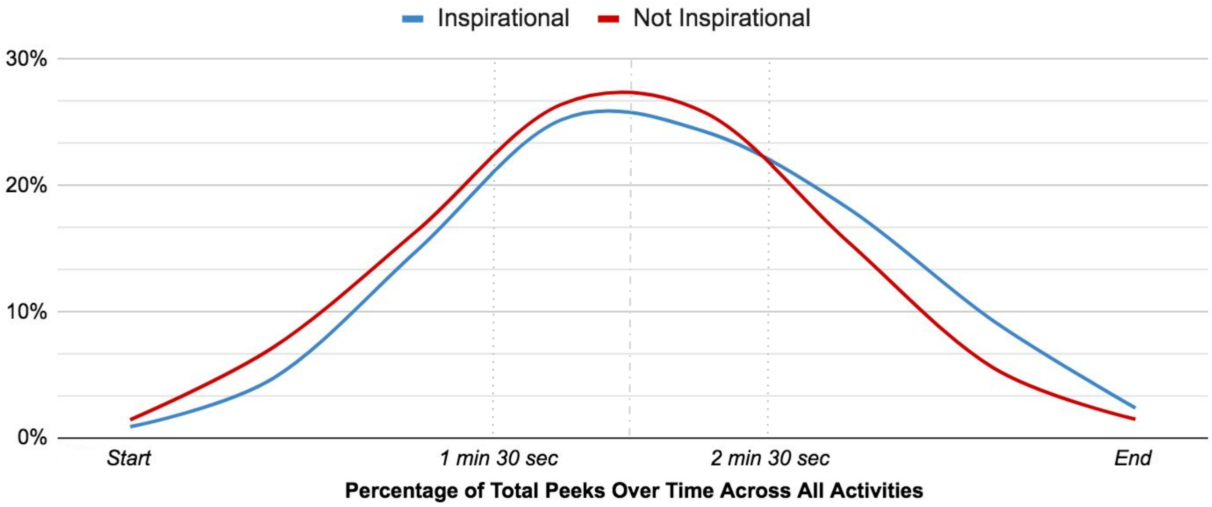


Figure 6.8: Distribution of peeks over time across both studies that led to and did not lead to design changes. Overall, students appear to sketch first and then peek later. While students peeked throughout each task, they most commonly peeked during the middle of each task, 68% of peeks occurred between 90–150 seconds after the task began.

In the large scale study, 90 students completed every task and survey, while 28 did so in the smaller setting. There were a total of 2,903 instances of peeking during the classroom study and 728 in the smaller setting. Overall, 58 students (64%) in the large classroom and 27 students (96%) in the smaller setting were inspired to change their sketch at least once. As shown in Figure 6.7, students took ideas back to their own sketches across a variety of tasks in both studies. Table 6.1 shows that peeking remained consistent across each task per study, demonstrating that students stayed engaged and continued to use Peek to seek ideas. Peeks in the final task of each study had the lowest chance of inspiring a design change. However, students peeked more frequently during these tasks, indicating a possible relationship between increased peeking and task difficulty.

Table 6.1 summarizes the peeking behavior of students using the Peek feature in Studies 1 and 2. Multiple Peeks were possible during a task. In the classroom study, peeking led to an intentional design change at some point during the task 17% of the time. In the subgroup in Study 1 that used Sketchy with Peek, this happened 18% of the time. The proportion of Peeks leading to a design change was similar between the two groups of users, for both sets of tasks were slightly different in what they asked for but retained the same format.

Digitally peeking occurred throughout both studies, with 68% of all peeks occurring around the middle of each task, see Figure 6.8. During the first minute of a task, it was 42% more likely for a sketch to be voted as non-inspirational. This indicates that a sketch might require more details in order for someone to find it inspirational. This cold-start problem begins to dissipate after the 1-minute mark. Across both studies, peeks leading to a reported design change occurred 16 seconds later than peeks that did not. A two-tailed t-test shows this 12% increase was statistically significant $t(3629) = 6.15$, $p < 0.01$, $d = 0.26$. These results remain consistent when analyzing Study 1 and Study 2 individually.

In Study 1, peeks resulting in an intended design change occurred 23 seconds later in the task. A two-tailed t-test show this 13% increase was statistically significant $t(726) = 2.93$, $p < 0.01$, $d = 0.28$. While in Study 2, peeks resulting in an intended design change occurred 13 seconds later in the task. A two-tailed t-test shows this 11% increase was also statistically significant $t(2901) = 5.84$, $p < 0.01$, $d = 0.30$. There is a cold start problem for finding sketches since Peek relies on sketches being created in real-time. Past research relied on prior examples and did not encounter this problem because they used completed sketches curated through pilot studies or online repositories [130, 205]. This idea is contrary to Sketchy's goal to allow anyone to create and run a sketching task without relying on prior sketches.

Sketchy allows students to see a peeked sketch update in real-time and the freedom to view the sketch for an unlimited amount of time. In both studies, a sketch that was changing while being peeked at did not affect whether a student would indicate if they were going to change their design. However, the length of time viewing a sketch while peeking did affect the student's behavior. In Study 1, students viewed sketches for 1.2 seconds longer when they indicated that they would make a design change. A two-tailed t-test shows this 35% increase was statistically significant $t(726) = 3.03$, $p < 0.01$, $d = 0.31$. In Study 2, students viewed sketches that they said would change their design for 0.4 seconds longer than a sketch they said would not. A two-tailed t-test shows this 14% increase was statistically significant $t(2901) = 2.32$, $p = 0.02$, $d = 0.11$. Results show that time spent viewing can predict whether someone will find a sketch inspirational. This discovery aligns with non-sketching research that focuses on either viewing time and increased interest, memorability, or attention [10, 25, 33]. This result could allow for the automatic rating of sketches as inspirational in real-time, as we discuss later in subsection 6.7.2.

6.6.4 Users Inspired by the More Experienced

In the pre-survey (see subsection 6.5.3), students self-reported their sketching and design experience on a 5-point Likert scale. These self-reported measures allow us to study whether the experience of peekers and creators influences digital peeking.

Comparing the experience of students who said they would change their sketch versus those who did not help to answer R2 by understanding why some students found sketches inspirational based on their self-reported experience. On average, students in Study 2 who said they would change their sketch when peeking had 17% less design experience and 10% less sketching experience than those who did not say they would change their sketch. A two-tailed t-test shows this decrease in design experience was statistically significant $t(2901) = -6.14$, $p < 0.01$, $d = 0.39$. A two-tailed t-test shows this decrease in sketching experience was statistically significant $t(2901) = -3.47$, $p < 0.01$, $d = 0.22$. These results may be unsurprising from a pedagogical standpoint but is a sign that users can gauge their sketching and design

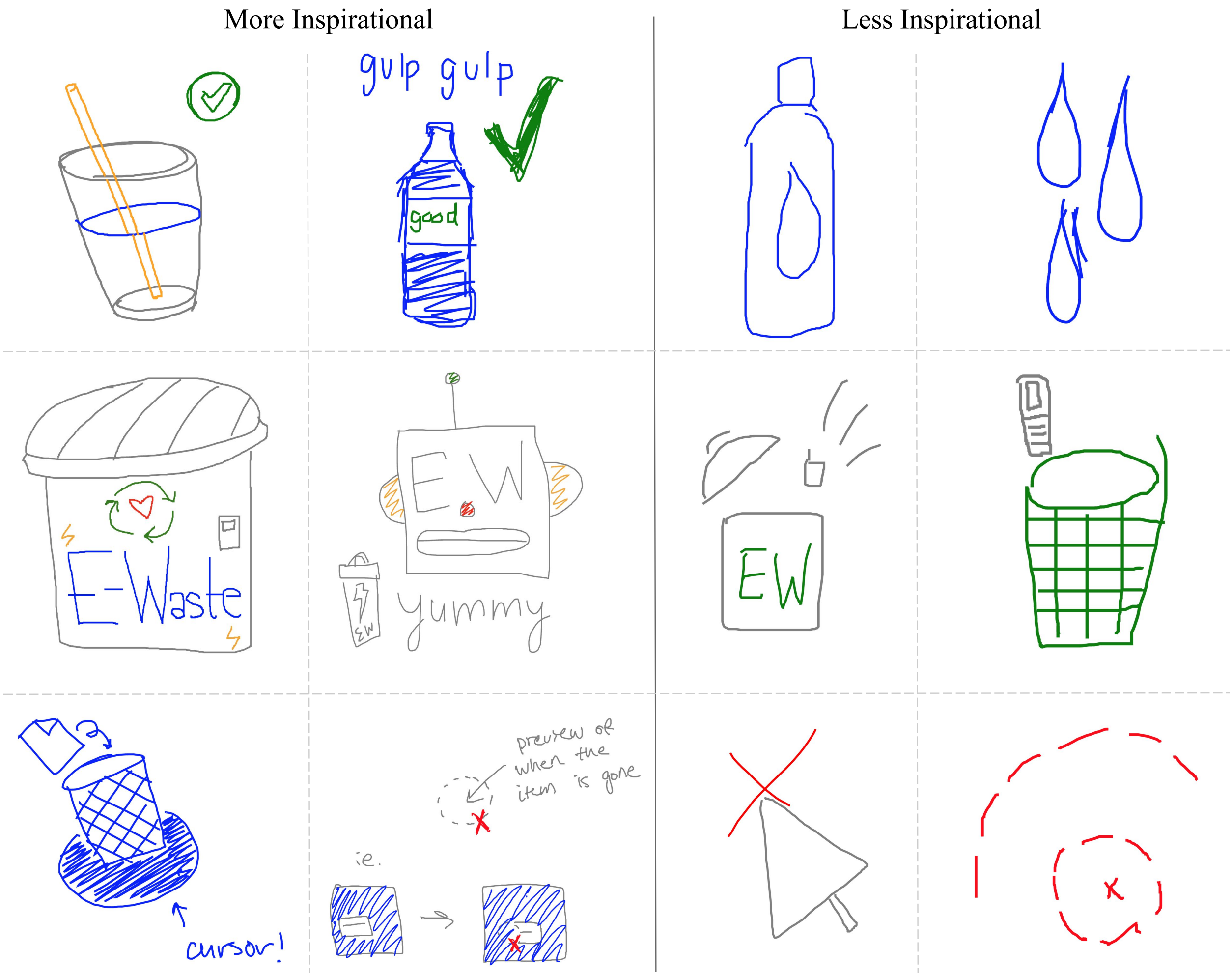


Figure 6.9: In-progress peeked sketches from Tasks 1–3 (top-bottom) from Study 2: The “more inspirational” sketches were often voted as inspirational and scored higher on each of the seven features *(box length, path length, stroke entropy, number of points, number of strokes, sum point differences, and sine end angle)*. The “less inspirational” sketches on the right received little to no votes that a student would change their sketching after peeking them. The inspirational sketches feature a variety of colors, well-spaced objects, long smooth strokes, and extra non-overlapping small strokes that add detail and clarity to the concept of the sketch.

confidence. These findings point to the possibility of being able to recommend a sketch to less experienced participants automatically.

Comparing the experience of the student who peeked versus the creator of the sketch can help answer R3 by understanding how differences between their self-reported experience measures might influence proposed sketch changes. In Study 2, students who intended to change their sketch after peeking had 13% less design experience and 6% less sketching experience on average than the creator of the peeked sketch. A paired two-tailed t-test shows this decrease in design experience was statistically significant $t(2902) = -3.62$, $p < 0.01$, $d = 0.31$, but the difference in sketching experience was not ($p = 0.14$).

While these findings show a possible beneficial relationship between a less experienced student peeking and a more experienced creator, what about the more experienced students who peeked? In Study 2, students who did not intend to change their sketch after peeking had 4% more design experience and 5% more sketching experience on average than the creator of the sketch they viewed. Separate paired two-tailed t-tests show these increases were statistically significant when comparing design experience $t(2902) = 3.33$, $p < 0.01$, $d = 0.10$; and sketching experience $t(2902) = 3.70$, $p < 0.01$, $d = 0.11$. These results show that peeking could benefit from pre-filtering sketches created by students' more experienced peers.

While we did not observe the same results in Study 1, future work could control for the task, class size, and distribution of sketching/design experience between groups to further study these occurrences. Future work could also develop individuated metrics to understand better how Peek could make recommendations that match a user's in-the-moment needs.

6.6.5 Peeking Influenced Student's Next Action to Contribute

Sketches that students found inspirational sketches were often more complete, more detailed, and carefully drawn than non-inspirational sketches, as seen in Figure 6.9. While the level of inspiration per sketch might differ between the students, their next action after digitally peeking can indicate how seeing a possible inspirational sketch can influence their next action. This analysis can help answer R2 for this chapter. After finishing peeking, they could perform one of eight possible actions listed in Table 6.2.

To compare the proportions of students performing one of the eight possible next actions versus one of the other actions, we computed two-tailed McNemar's tests with the continuity correction for all proportions. This test is appropriate to assess binary outcomes of repeated measures (in this case, the user actions).

			Additive			Subtractive	
Intended to change design after peeking	**Peek**	**View Stats**	**Change Color**	**Sketch**	**Redo**	**Undo**	**Clear**
Yes	73.8%	7.4%	10.8%	5.3%	0.0%	1.1%	1.6%
No	82.9%	7.2%	5.8%	3.0%	0.0%	0.5%	0.6%
Likelihood	0.9	1.0	1.9	1.7	0.0	2.3	2.8

Table 6.2: Likelihood to perform a specific action after digitally peeking, as observed in Study 1 (Peek Group) and Study 2 (Large Class). McNemar tests show all relationships with reported effect sizes were statistically significant $p < 0.01$. Notably, the largest differences were in actions that undid or cleared previous strokes, indicating students who intended to change their design were more likely to remove strokes from their design than those who did not.

Regardless of whether a student indicated they would change their sketch when peeking, the most common action was to peek again, as shown in Table 6.2. However, students across both studies who peeked and voted that they intended to change their sketch did not peek again 26% of the time, and students who indicated they would not change their design did not peek again 17% of the time. A McNemar test shows this 53% increase was statistically significant $\chi^2(1) = 2169.12$, $p < 0.01$.

While other actions were less common than peeking again, the differences in their occurrences indicate students are more likely to add or remove content from their sketch. If students added something new immediately after peeking, they would sketch again, change the color of their next stroke, or redo a previous action. After peeking, students who indicated they would change their sketch were 1.7 times more likely to sketch again as their next action and 1.9 times more likely to change the color of their next stroke than those who did not. Separate McNemar tests show these increases were statistically significant for both sketching $\chi^2(1) = 203.58$, $p < 0.01$ and changing the color of the next stroke $\chi^2(1) = 81.63$, $p < 0.01$.

If students removed something from their sketch immediately after peeking, they could undo the last stroke or clear their canvas to start over from nothing. After peeking, students who indicated they would change their sketch were 2.2 times more likely to undo their last stroke and 2.8 times more likely to clear their canvas than those who did not. Separate McNemar tests show these increases were statistically significant for undo $\chi^2(1) = 385.16$, $p < 0.01$; and clear $\chi^2(1) = 377.75$, $p < 0.01$. These subtractive actions show the biggest differences between actions following inspirational and non-inspirational peeks.

Students repeatedly peeking could indicate they want to explore different concepts. Streamlining digital peeking would allow users to consume more ideas quickly. With fewer repeated peeks, it might be possible to observe more non-peeking actions. Maybe students would perform more subtractive actions if Sketchy provided more sophisticated options to erase or undo specific strokes. This could allow for the analysis of what makes specific strokes undesirable after choosing to change their design. Alternatively, it might be quicker for someone to directly copy and paste strokes from a peeked sketch instead of attempting to imitate others' ideas.

6.7 Discussion

Peek relies on users concurrently generating ideas in real-time in a virtual space. This setting allows for these groups of users to simultaneously provide sketching examples to everyone else while individually working on their designs. This chapter's findings show Peek's effectiveness in real-time classroom activities to positively influence the likelihood someone contributes new ideas to their sketch after finding inspiration from their peers.

6.7.1 Sketchy in Real-Time Classroom Activities

Sketchy comprises several attributes making it particularly suited for real-time classroom activities: flexibility, student-to-student learning, and equitable access. As a design consideration informed by the preliminary observations, Peek's real-time nature gives instructors **flexibility** to define new tasks on an ad-hoc basis without additional preparation. No pre-selected inspirational content is necessary, as learning occurs through observing peers. This same flexibility applies to instruction during the task. The instructor can observe the students' progress and intervene by guiding as needed.

Our results reported that students sought conceptually different sketches for inspiration, particularly suited for Sketchy, where students have independent canvases. This independence allows students to continuously peek to find different concepts without worrying that someone else might alter their design. Peeking in this way fosters student-to-student learning in the form of imitation. As we found, less experienced users were more likely to plan changes when observing more experienced users. [141] similarly reported, “we observed students peeking on the canvases of other group members, even while still working on their own task.” Unlike prior work, Sketchy shows that retaining an independent canvas allows students to engage in **student-to-student learning** while making their own design decisions. Our results showed an increase in creativity measures, including the feeling of collaboration (from 2.9 to 3.5 with peeking). This shared progression is part of the educational experience, and as [141] also noticed, “[...] some teams [look] in on the efforts of other groups, in effect checking up on their progress.” Without a single best way to complete the task, creativity has a greater effect on the sketch.

The Sketchy application is accessible to students’ attendance and devices, which is vital in a changing educational environment. Its nature as a responsive web application allows for its use in virtual classrooms, in-person classrooms, or a mix of both (a hybrid classroom). Rather than physically peeking, students access that analogous feature as part of the application, offering **equitable access** to this part of the classroom experience. Whether a student is at home or in the classroom, the bring-your-own-device approach is simple to manage, with flexibility for users sketching with their fingers on a touchscreen or with laptop touchpads. Sketchy’s actual use in multiple classes shows that these three attributes– flexibility, student-to-student learning, and equitable access– make Sketchy a practical tool for real-time classroom activities.

6.7.2 A Smarter, More Personalized Peek Feature

Currently, the Peek feature selects sketches at random. This was influenced by feedback during pilot studies, where students were more interested in seeing many different examples.However, our results identify seven features from 2D stroke data, the timing of peeks, and self-reported measures of design and sketching experience that influence proposed sketch changes—our proxy for inspiration. We also notice that users want to see conceptually different-looking sketches, something we did not anticipate at the beginning of our study. This initial evidence can inform the development of a smarter, more personalized version of the Peek feature that can help answer the call to investigate “personalized inspirations” by [38] and [205]. We also observed how users imitated ideas they repeatedly saw, despite never indicating an intention to imitate a specific sketch they peeked. This observation leads to the idea that a personalized feature should still integrate a level of randomness in selecting examples to show users.

6.8 Limitations and Additional Future Work

Sketchy enables us to explore the concept of design inspiration in the context of viewing another person's sketch during co-occurring real-time design tasks. The participants in our studies had exposure to Sketchy for a limited period of around 20 minutes. Despite evidence that creativity measures increased with the Peek feature's introduction, it is not clear if this was due to the novelty of the feature at the time. Sketchy does not necessarily deliver a tangible approach to improving users' creativity over time, as this would require a consideration of what happens when someone repeatedly uses a tool like Sketchy.

One area for practical, creative improvement is to determine which peeked sketches cause design changes. It is exceptionally challenging to automatically quantify if a user copies specific strokes from the peeked sketch. They might not have the skill to recreate a stroke, or they might only imitate part of an idea, such as size, placement, or orientation. This is why our work relies on a proxy question and studying their next action after peeking. In a more direct method, users could copy-paste strokes directly from a peeked sketch. Then we could study how users integrate and iterate these "copied" strokes through a task. Future analysis could also measure the proportion of strokes or points users directly copy from other sketches.

6.9 Conclusion

This chapter introduces Sketchy's digital Peek functionality to group design settings, and we found that collaboration, facilitated by the Peek feature, is key to increasing users' overall creativity and satisfaction with their final sketch. Users drawing with the Peek feature displayed statistically significant increases in the overall CSI score and three components: Collaboration, Expressiveness, and Effort/Reward Tradeoff. These results demonstrate that Peek better supports individual creativity and fulfills its potential to increase collaboration by translating physical behaviors into the digital realm.

Sketchy's unpaid contributors (i.e., users) were motivated by contributing sketches and engaging with their peers. One observation during the activities is how people ask their peers from across the room if a particular sketch was theirs. This sense of collaboration helped motivate quality contributions through sketches. While they did not directly collaborate, the more experienced users were able to contribute their specialized knowledge through higher-quality sketches. Thus, the Peek feature allowed them to simultaneously share their contributions with their peers or someone in the class they did not know. Sketchy's unpaid contributors were more engaged as the effort to contribute was reduced, and the sense of collaboration was heightened. When contributing high-quality sketches, this helped their peers, thus increasing the number of people who stayed and completed the sketching tasks (i.e., conversion rate). Overall, Sketchy's system design provides several factors that intrinsically motivate a group of unpaid contributors to provide high-quality contributions without the need to provide direct monetary compensation for their efforts.

Sketchy supports a fast perpetual cycle of data creation and improvement, which allows this public data system to create and maintain a set of potentially inspirational sketches in real time. Sketchy's design was influenced by real-world observations of people performing sketching tasks. We found that Peek increases engagement and contributions within short 4-minute sketching tasks. Users generally found sketches inspirational when they were more detailed, complete, carefully drawn, or drawn by someone more experienced.

The development of Sketchy and the Peek feature serves as a turning point in this dissertation's definition of a public data system and how to motivate users in the wild. The lessons learned from Sketchy's iterative development process and user studies will help influence a redesign of Drafty to create a complete public data system. For example, providing users agency in their interactions and methods to further engage with the evolving information can increase engagements that benefit the individual user and the quality of the system's information. The system should not nudge users to contribute but instead provide features they can freely interact with to create a more engaging experience. By seeing an inspirational sketch using Peek, users were more likely to contribute information in the form of sketches back to the system. These lessons can influence a redesign of a public data system, like Drafty, to provide users with more agency over their interactions and to create methods for users to interact with data to motivate continued engagement and contributions.

Chapter 7

Attract & Engage Users with Insights to Maintain Evolving Information

This chapter presents a longitudinal naturalistic study evaluating a new version of Drafty with additions such as Computer Science Open Rankings (meta-ranking for Computer Science departments) and Databaits (an API to generate insights from sentence templates and aggregate statistics using tabular data). This chapter studies how these additions and other sourcing methods, such as asking people on Twitter or posting on forums, attract visitors to Drafty and their accuracy per visit, the likelihood they contribute, edits per visit, and their average engagement score per visit. This chapter is part of a future journal paper. I am the first author and responsible for all aspects of user studies, system design, implementation, analysis, and almost all of the writing. As part of the contributions of this chapter, the new version of Drafty, CS Open Rankings, Databaits, and the tabular dataset of Computer Science professors are publicly available[1].

7.1 Introduction

There are positive examples of small communities maintaining small public datasets in spreadsheets (contrary to the large datasets in Wikipedia and Wikidata)—for example, the successful efforts within our public web application "Drafty" to maintain Computer Science professors' academic profiles over several years (see chapters 3 and 5). Also, at the onset of the COVID-19 pandemic, people made unpaid contributions (i.e., edits) to maintain a spreadsheet of university campus closures and migrations to online learning. These observations show that spreadsheets, used by over 750 million people in Excel alone, are ideal for studying how people maintain evolving data. One factor

[1] Available at http://drafty.cs.brown.edu.

remains constant between Drafty and the COVID-19 pandemic school datasets. Individuals made small and accurate contributions and often sought insights from the spreadsheet's data. Therefore, to maintain this data as it changes, it is necessary to develop a mechanism to extract insights from the spreadsheet's data, share these insights with the public, and continuously share the benefits of the public data with the community that supports it.

A public data system, such as Drafty, requires a constant flow of people to visit, review data, and make contributions to maintain its evolving information. This dissertation's preliminary work shows unpaid contributors can make accurate edits across data types requiring varying levels of domain-specific knowledge (see chapters 3 and 5). This chapter focuses on methods and features to engage users within Drafty and to study new sources derived from Drafty's data to attract visitors who make accurate contributions. As part of this effort, we developed a new version of Drafty and several new systems that rely on its data, such as Computer Science Open Rankings, the Databaits API, and Computer Science Open Data.

Diversifying the approach to attract visitors and studying the increase in unpaid contributions mirrors the recommendation to study natural behavior in social computing systems [19]. Understanding which visitors to a public system will contribute is an open research question [242]; simultaneously, evaluating multiple methods to attract visitors will be vital to understanding possible user motivations in the wild. This analysis can split visitors by the resource they arrived from, for example, automatic insights (CS Open Rankings and the Databaits API), manually shared static insights and messages (CS Open Data, emails, Twitter), and organic visitors arriving from search traffic or other methods we did not control.

These chapters demonstrate while asking users to fix data matching their interests produces highly accurate edits, users only completed the request 9% of the time. Thus, Drafty requires a mechanism to motivate contributions while not interrupting users during their natural information-seeking tasks. Previous research shows how to craft messages to nudge users to take action to voluntarily contribute their time to causes [53, 200]. However, nudging can be considered manipulative, and these prior efforts are short-lived. When researchers deployed systems in the wild, they found it imperative to balance users' trust and agency with the system [60, 195]. Therefore, users should be free to engage with a system like in peer production, but the tasks they perform within the system are directed and specific similar to crowdsourcing [16]. Chapter 6 also shows initial evidence during the development of the public data system, Sketchy that providing users agency within the system's confines can increase good engagement and beneficial contributions. So, within Drafty, what methods and mechanisms can be studied in a naturalistic scenario to attract and engage a group of anonymous users sharing common interests to make unpaid contributions to evolving data?

The research asks the following questions:

1. What methods to attract anonymous visitors to a public data system yield more edits per visit, accuracy per edit per visit, and more engagement?

2. How does providing automatic insights (from the Databaits API) within Drafty encourage anonymous visitors to engage within the system and make accurate edits per visit?

This research shows that integrating insights dynamically generated from Drafty's tabular can be used to attract visitors who make more accurate contributions. Notably, their accuracy per visit is similar to directly asking friends or members of the Computer Science community to review and edit Drafty's data. These sources integrating automatic insights require lower long-term effort to maintain and run compared to personally asking people to contribute to public data systems like Drafty.

7.2 Related Work

Understanding how to translate tabular data into text can be a complex task. This chapter develops the CS Open Rankings and the Databaits API to accomplish this task automatically. Prior work implements some ideas about translating tabular and structured data into human-readable insights.

7.2.1 Using Natural Language Descriptions of Data to Engage Visitors

Producing natural language from tabular data is a rich research area. There have been many systems aimed at completing natural language generation tasks—where natural language text is produced from non-textual data like tables. Most natural language generation systems so far have focused on using various types of machine learning models. Langkilde and Knight used a statistical language n-gram based system. Sutskever et al used a neural net model. Zhang's system, also through a neural net model, [240] generates representative statements about the inputted tables. This task is relatively similar to Databaits, except the statements produced by Databaits are facts that draw on specific cells within the data, while Zhang's statements are descriptions of the table in their entirety. Parikh proposed a similar task of generating a single sentence based on a table and various highlighted columns, similar to the methodology behind Databaits generation. Mahapatra [148] proposes a knowledge-light non-rule-based system to create relevant natural language phrases from input data.

Previous work has used a template-based system to semi-automatically generate fun facts based on Wikipedia tables by Korn et al. [126]. The Databaits API and their system aim to produce interesting factoids from tables. While their study focused primarily on identifying the tables within an article which would most likely produce interesting facts, the Databaits API focuses on producing varied, interesting insights from one large tabular dataset. Both systems require manually creating sentence templates to regenerate and show users different insights. Adeyanju [2] found that knowledge-intensive systems, such as those that use templates or rules, typically perform better when compared to knowledge-light systems per human evaluators.

The Databaits API human-consumable insights are derived from Drafty's large tabular dataset. By focusing on one large dataset of thousands of rows, our efforts go beyond prior research that links small related tables on Wikipedia [72]. Building on these ideas of fact-based sentences, SuggestBot shows how to structure these sentences to maximize their ability to engage people [53]. They structure sentences using a "High-Involvement style." Other research results show this style increases long-term retention among paid crowdworkers participating in conversational micro-tasks [178]. Our Databaits API generates facts, or automatic insights, by building on these ideas. Similar to this prior work, the goal of the Databaits API is to automatically create and structure these facts from tabular data to attract, engage, and motivate people to visit and contribute to Drafty.

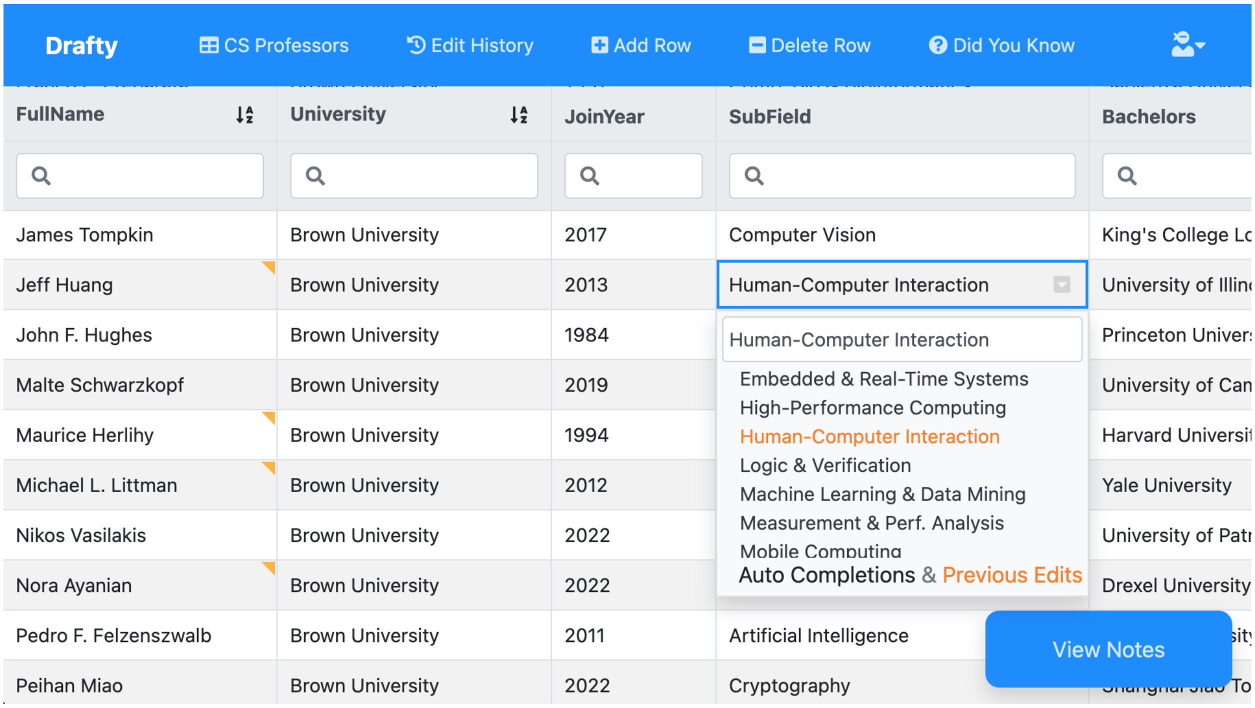
7.3 A Public Platform for Maintaining Tabular Data: Drafty, CS Open Rankings, and the Databaits API

This current chapter and Chapter 8 feature multiple studies conducted from 2021 to 2023. To conduct these studies, we developed a new version of the interactive spreadsheet web application, "Drafty", as the core part of our public platform to maintain tabular data. Drafty is a public data system that supports free access and open edits to tabular data. During these studies, Drafty hosts a crowdsourced dataset of tenure-track Computer Science profiles from top Universities in the US and Canada. Each profile consists of their full name, the university they are employed at, the year they joined that university as a tenure-track professor, their research area of expertise (i.e., subfield), and the name of the institutions that awarded their Bachelor's and Doctorate degrees. During this three year period, we also developed extensions that use this CS professor data to create rankings and insights: CS Open Rankings and the Databaits API. CS Open Rankings is a meta-ranker of Computer Science departments in the US and Canada that includes four rankings users can toggle to create their own ranking. The Databaits API uses Drafty tabular data to generate insights by using the column names and values to generate statistics and plug them into sentence templates. In 2022, Drafty integrated the Databaits API within its spreadsheet interface as the new feature "Did You Know." This allows anyone to freely see and create insights from its Computer Science professor profiles using the Databaits API.

The following sections describe how we built Drafty's spreadsheet, CS Open Rankings, the Databaits API, and the "Did You Know" feature. These systems and extensions create a public platform to attract, engage, and incentivize regular everyday people to visit, engage, contribute, and maintain the CS professor profile source data over time.

7.3.1 Developing a New Drafty, an Interactive Web-based Spreadsheet for Data Maintenance and Discovery

"Organic" visitors can freely visit the Drafty web application through various means, such as search engines, typing the URL in the browser, or other people sharing a link to Drafty's public web app. Chapters 3 and 5 use the original version of Drafty to conduct longitudinal studies. This chapter features a new version of Drafty developed from the ground up to attract more organic visitors; see Figure 7.1 for the updated interface.



Drafty
CS Professors
Edit History
Add Row
Delete Row
Did You Know
Human-Computer Interaction
Embedded & Real-Time Systems
High-Performance Computing
Human-Computer Interaction
Logic & Verification
Machine Learning & Data Mining
Measurement & Perf. Analysis
Mobile Computing
Auto Completions & Previous Edits
View Notes

FullName	University	JoinYear	SubField	Bachelors
James Tompkin	Brown University	2017	Computer Vision	King's College Lc
Jeff Huang	Brown University	2013	Human-Computer Interaction	University of Illin
John F. Hughes	Brown University	1984		Princeton Univer:
Malte Schwarzkopf	Brown University	2019		University of Can
Maurice Herlihy	Brown University	1994		Harvard Universit
Michael L. Littman	Brown University	2012		Yale University
Nikos Vasilakis	Brown University	2022		University of Patr
Nora Ayanian	Brown University	2022		Drexel University
Pedro F. Felzenszwalb	Brown University	2011	Artificial Intelligence	sity
Peihan Miao	Brown University	2022	Cryptography	Shanghai Jiao To

Figure 7.1: A new version of Drafty, a new publicly editable spreadsheet developed from the ground up to support low-effort edits, edit history, infinite scrolling, and thousands of rows of search-indexed data.

Drafty's prior version performed poorly in search engine results. While it could present thousands of rows of data to users while they scrolled and searched a tabular dataset, these results were not being search indexed properly. This is because new rows of data were dynamically retrieved from the server based on user interactions, such as scrolling. This new version of Drafty needed to handle infinite data client-side while providing a similar user experience with scrolling and search actions. We developed the pure client-side javascript library PolyMu[2] to handle and present hundreds of thousands of rows of data to the user that is also search-indexed. The spreadsheet is statically generated server-side every couple of minutes. The data for the table is stored in an HTML template element. The HTML template element has two benefits. Any data within the template element will be search indexed. And HTML inside the template element will not be rendered. This enables the page to load quickly without the size of the table affecting page load speeds.

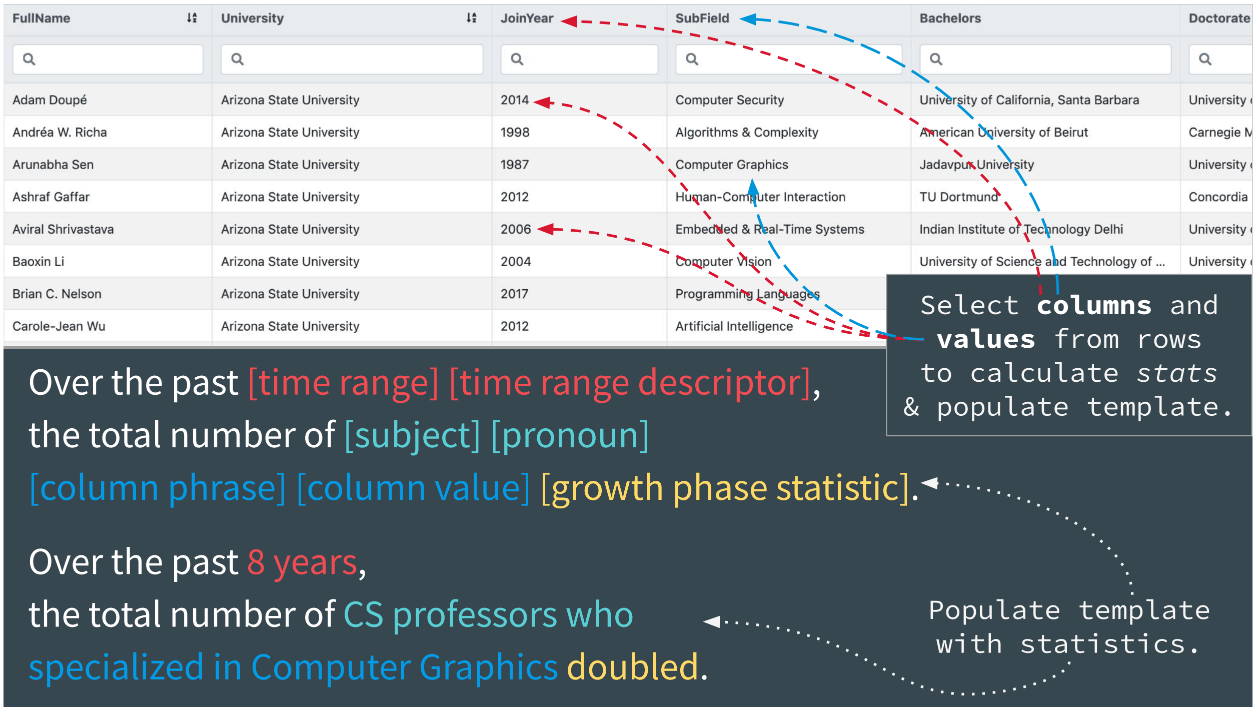
[2] https://github.com/pengzhengyi/polymu

PolyMu stores the table from the HTML template element in memory. Then, PolyMu dynamically renders the correct row of data based on a user's scroll position, column sorting, and column search inputs.

A key design focus for this new version of Drafty is to reduce the effort to edit cells, add new rows, and delete existing rows of data. Decreasing the time and effort to edit data should increase visitors' motivation to contribute by reducing a barrier. Also, this should decrease the time, and subsequent monetary cost, when future paid crowdworkers edit data in Drafty. When editing a single cell, a drop-down with autocomplete appears directly below the cell. This interface provides suggested values and highlights any previously contributed values for that row of data in orange. The previous version of Drafty featured a lengthy process to add a new row of data by guiding a visitor through a "wizard" dialogue process that featured several steps. Chapters 3 and 5 show that prior visitors were less likely to submit new rows of data due to the effort required to move through this step-by-step interface. This new version of Drafty allows visitors to add a new row of data featuring an inline form presented within the spreadsheet itself. When deleting a row, a visitor must add a reason they are removing the row. This will help data curators check why a row was deleted. For example, it could be a duplicate or the professor could have retired.

7.3.2 Databaits: Automatically Extracting Insights from Tabular Data

Databaits, developed and deployed as a restful API, creates insights from tabular data. Each Databait type (10 total) is a different sentence template merged with values and computed statistics derived from any tabular dataset. For example, Figure 7.2 shows how Databaits generates an insight using Drafty's Computer Science dataset.



FullName	University	JoinYear	SubField	Bachelors	Doctorate
Adam Doupé	Arizona State University	2014	Computer Security	University of California, Santa Barbara	University
Andréa W. Richa	Arizona State University	1998	Algorithms & Complexity	American University of Beirut	Carnegie M
Arunabha Sen	Arizona State University	1987	Computer Graphics	Jadavpur University	University
Ashraf Gaffar	Arizona State University	2012	Human-Computer Interaction	TU Dortmund	Concordia
Aviral Shrivastava	Arizona State University	2006	Embedded & Real-Time Systems	Indian Institute of Technology Delhi	University
Baoxin Li	Arizona State University	2004	Computer Vision	University of Science and Technology of ...	University
Brian C. Nelson	Arizona State University	2017	Programming Languages		
Carole-Jean Wu	Arizona State University	2012	Artificial Intelligence		

Select **columns** and **values** from rows to calculate *stats* & populate template.

Over the past [time range] [time range descriptor], the total number of [subject] [pronoun] [column phrase] [column value] [growth phase statistic].

Over the past 8 years, the total number of CS professors who specialized in Computer Graphics doubled.

Populate template with statistics.

Figure 7.2: Insights from a CSV are created by providing column names and column values as part of the input to the Databaits API. The Databaits API computes statistics using these inputs and merges the statistics with a randomly or intentionally chosen sentence template. This creation method for insights is designed to work best with tabular data (i.e., CSVs) containing thousands of rows of data.

This section describes how the Databaits API creates insights from tabular data, a preliminary survey study to analyze people's impressions and reactions to these types of insights, and finally, how the Databaits API is integrated into Drafty to create the "Did You Know" feature. "Did You Know" allows users to freely see and generate insights in Drafty and tweet them to Drafty's Twitter account. The "Did You Know" feature is directly inspired by Sketchy's feature from Chapter 6, providing users the freedom to generate random examples as insights from the source data.

Databaits API: Design Considerations for Creating Insights from Tabular Data

We developed the Databaits API to generate insights in two main steps. First, it uses values from rows and labels from columns from a tabular dataset to compute aggregate statistics. Second, it merges these values, labels, and statistics into sentence templates. The Databaits API can create ten different kinds of insights or Databait types. Each Databait type has a unique statistic and a unique sentence template. The behavior of generating statistics and parts of the sentence templates can be modified using different inputs. The following paragraphs explain this in detail.

Databaits API: Creating Insights Overiew First, the Databaits API takes in a list of column names and/or column values the user has interacted with. For example, if a user is viewing a table about video games and clicks on a cell in the "Publisher" column with the value "Nintendo," then our system would take the column "Publisher" and column value “Nintendo” as inputs. Then the Databaits API calculates statistics about these column names and/or column values. These statistics might be the average value of a column or the rate of change of a column value over time.

In total, the Databaits API can calculate 10 different types of insights. Each type has a unique statistic and associated sentence template. See the appendix for the full pseudocode of each Databait type.

For example, Databait Type 1 measures how the frequency of a column value changes over time. The following would represent a Type 1 Databait: “Over the past 25 years, the total number of games that were published by Sony Computer Entertainment increased 97 times.” Another example of a Type 1 Databait is: “Over the past 10 years, the total number of games that were designed for Wii increased by 53 percent.”

While these two sentences feature different column values (the former Sony and the latter Wii), they both measure the growth of one column value over time. Therefore, these Databaits are both of Type 1. On the other hand, Databait Type 2 compares the frequency of one column value to that of another. For example, a Type 2 Databait might be that “Four times as many games were designed for Wii than for DS in the past 5 years.” This statistic is structurally different from that used in Type 1, in that it compares two different column values instead of focusing on one. Therefore, this second set of insights are designated as a different type.

Databaits API: Configuring System Behaviors to Create Insights When the Databaits API creates an insight, it can take in as optional inputs the type of Databait to create, multiple column names, multiple values per column, and any of the other configuration parameters below:

- subject: the entity being described by each row in the table (e.g., “CS professors”).
- pronoun: the pronoun that represents the subject (e.g., “who”).
- column names(s): Often the column name, that describes a column’s information (e.g., “University”).
- column phrase(s): a term or phrase that describes what the column’s name does (e.g., “specialized in” for Subfield).
- column value(s): a unique set of values from a column.
- comparison statistic descriptor: a phrase that describes statistics that compare one column value to another (e.g., “twice as many”).
- growth phase statistic descriptor: a phrase that describes statistics that track the change over time of one column value (e.g., “doubled”).
- time range: the distance between the minimum and maximum for a time-based column value.
- time range descriptor: a unit of measurement for the time range (e.g., “years”).

- historical event(s): historical events and they year/s they took place for Databait Type 7 (e.g., “the iPhone was released” and “2007”).

These can be provided by passing a configuration file, or the Databaits API can select any of these at random.

For example, Figure 7.2’s sentence template consists of static and variable parts. The variable parts are colored for distinction. These parts change based on which column names or column values the Databaits API used to generate each statistic. The static portions describe information inherent to the Databait type. These are shared across Databaits of the same type.

By allowing this type of configuration per dataset it expands the variety of data types and datasets that can be plugged into the Databaits API to generate insights.

Databaits API: Calculating Statistics for Databaits When the Databaits API is called, it either uses the pre-defined Databait type provided in the call or randomly selects a suitable Databait type to create that can handle the number of columns and values provided. If it cannot compute a statistic given the provided inputs, the Databaits API will select random column names and values and create a random Databait type. After successfully creating the aggregate statistic, it will attempt to plug it into the sentence template. If this last step is successful, the Databaits API will return the final insight through its restful API.

To ensure the creation of diverse insights and statistics, the Databaits API does not attempt to compute statistics with the largest effect sizes. For example, in the Computer Science professor dataset, every insight comparing the number of faculty per department would likely select Carnegie Mellon University because they have two to ten as many professors as other departments. Future work could optimize these effect sizes to find the right balance between variety and usefulness per insight.

Databaits API: Generating Sentences Once a statistic is calculated, it is converted into an English sentence. Each Databait type has its own unique template consisting of static and variable parts. The variable parts change depending on which inputs were used to generate the statistic.

Some variables, such as the subject and time range, can be plugged into sentence templates without modification. While, column names such as “University”, “SubField”, or perhaps something more syntactically complicated such as “global_sales”, must be translated into plain English before they can blend into sentences. To do so, the user specifies a dictionary of *column descriptors*, which maps each column name to an English phrase that describes its function. For example, the “SubField” column could be mapped to the phrase “specialize in.” Combined with the *subject* and *pronoun* variables, column descriptors create a natural phrase that depicts the kind of data represented by each column. In this case, the full phrase would be “CS professors who specialize in” followed by a column value of the “SubField” column such as “Human-Computer Interaction.”

Statistics are also converted into plain English in order to make information more digestible. If the insight is comparing two column values, the figure will be translated into a *comparison statistic descriptor*, such as "twice as many" or "nearly 10 times that of." Suppose the insight is following the increase or decrease of a value over time. In that case, the figure will be converted into a *change over time statistic descriptor*, such as "doubled" or "grew nearly 10 times," followed by a *time range* variable, such as "over the past five years."

Databait Types (1–10): Sentence Templates and Examples

The list below contains a quick summary of each Databait type. The full pseudo-code for all 10 Databait types is below the following list:

1. Databait Type 1 describes the change over time of a single column value.
2. Databait Type 2 compares the total frequency of two column values from one column within a time frame.
3. Databait Type 3 measures the change over time in the frequency of entries that have a certain column value for one column, and another column value for a second column. Essentially, it follows entries that have two specified features instead of one.
4. Databait Type 4 compares the frequency of entries that share one column value for one column, but have two different column values for a second column.
5. Databait Type 5 compares the frequency of a column value to the average frequency of all column values in that column.
6. Databait Type 6 describes the proportion of entries that have a specific column value for a given column.
7. Databait Type 7 measures the change over time in the number of entries with a column value, starting from a user-specified year in which a historic event took place. The configuration file provides historical events and the years they took place.
8. Databait Type 8 measures the proportion of entries that share a column value in two different columns.
9. Databait Type 9 measures the proportion of all entries that hold a specific column value for a column, for two different time points.
10. Databait Type 10 compares one column value that is represented by more entries now than before, with one that is represented by less.

Databait 1 Databait 1 describes the change over time of a single column value.

- Template: Over the past *time range* years, the total number of *subject* / *pronoun* / *column descriptor* / *column value* / *change over time statistic descriptor*.
- Sample: Over the past *25 years*, the total number of *CS professors* / *who* / *specialized in* / *Human-Computer Interaction* / *more than quadrupled*.

```
Input: $i \leftarrow$ column value, $c \leftarrow$ column name
Output: $bestRange$, the time range that produces the greatest rate of change in the number of values with $i$ for $c$ compared to the current year, and $maxChange$, the rate of change
  $D \leftarrow$ rows in dataset with $i$ for $c$
  $T \leftarrow$ current year
  $maxChange \leftarrow 0$
  $bestRange \leftarrow 0$
  for $t \in \{T, T-5, T-10, ...\}$ do
    $change \leftarrow$ rate of change of number of entries in $D$ from year $t$ to now
    if $change > maxChange$ then
      $maxChange \leftarrow change$
      $bestRange \leftarrow t$
    end if
  end for
```

Databait 2

Databait 2 compares the total frequency of two column values from one column within a time frame.

- Template: *Comparison statistic phrase* / *subject* / *column descriptor* / *first column value* than by *second column value* in the past *time range*.
- Sample: *6 times as many* / *CS professors* / *were hired by* / *Carnegie Mellon University* than by *Brown University* in the past *25 years*.

```
Input: $i_1 \leftarrow$ first column value, $i_2 \leftarrow$ second column value, $c \leftarrow$ column name
Output: $bestRange$, the time range that produces the greatest percentage difference in the number of values with $i1$ and $i_2$, and $maxDiff$, the difference
  $D_1 \leftarrow$ rows in dataset with $i_1$ for $c$
  $D_2 \leftarrow$ rows in dataset with $i_2$ for $c$
  $T \leftarrow$ current year
```

```
$maxDiff \leftarrow 0$
$bestRange \leftarrow 0$
for $t \in \{T, T-5, T-10, ...\}$ do
    $diff \leftarrow$ percentage difference between total number of entries in $D_1$ and that of $D_2$ from year $t$ to now
    if $diff$ ¿ $maxDiff$ then
        $maxDiff \leftarrow diff$
        $bestRange \leftarrow t$
    end if
end for
```

Databait 3

Databait 3 measures the change over time in the frequency of entries that have a certain column value for one column, and another column value for a second column. Essentially, it follows entries that have two specified features instead of one.

- Template: Over the past *time range*, the total number of *subject* / *pronoun* / *first column descriptor* / *first column value* and *second column descriptor* / *second column value* / *change over time statistic descriptor*.
- Sample: Over the past *25 years*, the total number of *CS professors* / *who* / *were hired by* / *Carnegie Mellon University* and *specialized in* / *Artificial Intelligence* / *increased 7 times*.

Input: $i_1 \leftarrow$ first column value, $c_1 \leftarrow$ first column name, $i_2 \leftarrow$ second column value, $c_2 \leftarrow$ second column name

Output: *bestRange*, the time range that produces the greatest rate of change in the number of values with i_1 for c_1 and i_2 for c_2 compared to the current year, and *maxChange*, the rate of change

```
$D \leftarrow$ rows in dataset with $i_1$ for $c_1$ and $i_2$ for $c_2$
$T \leftarrow$ current year
$maxChange \leftarrow 0$
$bestRange \leftarrow 0$
for $t \in \{T, T-5, T-10, ...\}$ do
    $change \leftarrow$ rate of change of number of entries in $D$ from year $t$ to now
    if $change > maxChange$ then
        $maxChange \leftarrow change$
        $bestRange \leftarrow t$
    end if
end for
```

Databait 4

Databait 4 compares the frequency of entries that share one column value for one column, but have two different column values for a second column.

- Template: *Comparison statistic descriptor / subject / pronoun / first column descriptor / shared column value / second column descriptor / first column value* than by *second column value* in the past *time range*.
- Sample: *Twice as many / CS professors / who / specialized in / Databases / were hired by / Brown University* than by *Carnegie Mellon University* in the past *20 years*.

```
Input: $s \leftarrow$ shared column value, $c_1 \leftarrow$ first column name, $i_1 \leftarrow$ first unique column value, $i_2 \leftarrow$ second unique column value, $c_2 \leftarrow$ second column name
Output: $bestRange$, the time range that produces the greatest percentage difference in the number of values with $s$ in $c_1$ and $i_1$ in $c_2$ vs. $s$ in $c_1$ and $i_2$ in $c_2$, and $maxDiff$, the difference
  $D_1 \leftarrow$ rows in dataset with $s$ in $c_1$ and $i_1$ for $c_2$
  $D_2 \leftarrow$ rows in dataset with $s$ in $c_1$ and $i_2$ for $c_2$
  $T \leftarrow$ current year
  $maxDiff \leftarrow 0$
  $bestRange \leftarrow 0$
  for $t \in \{T, T-5, T-10, ...\}$ do
    $diff \leftarrow$ percentage difference between total number of entries in $D_1$ and that of $D_2$ from year $t$ to now
    if $diff$ ¿ $maxDiff$ then
      $maxDiff \leftarrow diff$
      $bestRange \leftarrow t$
    end if
  end for
```

Databait 5

Databait 5 compares the frequency of a column value to the average frequency of all column values in that column.

- Template: *Comparison statistic descriptor / subject / column descriptor / column value* than in the average *column name* in the past *time range*.

- Sample: *5 times as many* / *CS professors* / *specialized in* / *Machine Learning and Data Mining* than in the average *SubField* in the past *25 years*.

```
Input: $c \leftarrow$ column name
Output: bestRange, the time range that produces the greatest percentage difference in the number of values with the mode column value for $c$ and the average, and maxDiff, the difference
  $maxDiff \leftarrow 0$
  $bestRange \leftarrow 0$
  $mode \leftarrow$ None
  $T \leftarrow$ current year
  for $t \in \{T, T-5, T-10, ...\}$ do
    $diff \leftarrow$ percentage difference between total number of entries in $D_1$ and that of $D_2$ from year $t$ to now
    if diff ¿ maxDiff then
      $maxDiff \leftarrow diff$
      $bestRange \leftarrow t$
    end if
  end for
```

Databait 6

Databait 6 describes the proportion of entries which have a specific column value for a given column.

- Template: *Raw statistic* percent of all *subject* from the past *time range* / *column descriptor* / *column value*.

- Sample: *16* percent of all *new CS professors* from the past *5 years* / *specialized in* / *Machine Learning and Data Mining*.

```
Input: $c \leftarrow$ column name
Output: bestRange, the time range for which the mode of $c$ holds the largest proportion of $c$, bestMode, the mode for that time range, and maxProportion, the percentage of values in $c$ that equal bestMode for that time range
  $maxProportion \leftarrow 0$
  $bestRange \leftarrow 0$
  $bestMode \leftarrow$ None
  $T \leftarrow$ current year
  for $t \in \{T, T-5, T-10, ...\}$ do
    $mode \leftarrow$ mode in $c$ from time $t$ to now
    $proportion \leftarrow$ percentage of values in $c$ from time $t$ to now that equal $mode$
    if $proportion$ ¿ $maxProportion$ then
        $maxProportion \leftarrow proportion$
        $bestRange \leftarrow t$
        $bestMode \leftarrow mode$
    end if
end for
```

Databait 7

Databait 7 measures the change over time in the number of entries with a column value, starting from a user-specified year in which a historic event took place. The configuration file provides history events and the years they took place.

- Template: The total number of *subject* / *pronoun* / *column descriptor* / *column value* / *change over time statistic descriptor* since *user-specified historic event*.
- Sample: The total number of *CS professors* / *who* / *specialized in* / *Artificial Intelligence* / *has nearly quadrupled* since *1997, when IBM's Deep Blue beat Garry Kasparov*.

```
Input: $c \leftarrow$ column name, $i \leftarrow$ column value, $t_* \leftarrow$ time at which special event occured
Output: $change$, the rate of change in the number of values in $c$ that equal $i$, from time $t_*$ to now
    $T \leftarrow$ current year
    $n_1 \leftarrow$ number of rows in dataset with $i$ for $c$ and $t_*$ for the time column
    $n_2 \leftarrow$ number of rows in dataset with $i$ for $c$ and $T$ for the time column
    $change \leftarrow \frac{n_2 - n_1}{T - t_*}$
```

Databait 8

Databait 8 measures the proportion of entries that share a column value in two different columns.

- Template: *Raw statistic* percent of *subject* / *first column descriptor* and *second column descriptor* the same *column name*.
- Sample: *13* percent of *CS professors* / *got their bachelor's degrees from* and *got their doctorate degrees from* the same *university*.

```
Input: $s \leftarrow$ shared column value, $c_1 \leftarrow$ first column name, $c_2 \leftarrow$ second column name
```

```
Output: bestRange, the time range that produces the greatest percentage of total entries that have $s$ in $c_1$ and $c_2$, and
    maxProportion, the percentage
    $T \leftarrow$ current year
    $maxProportion \leftarrow 0$
    $bestRange \leftarrow 0$
    for $t \in \{T, T-5, T-10, ...\}$ do
        $total \leftarrow$ total number of entries from time $t$ to now
        $n \leftarrow$ total number of entries from time $t$ to now with $s$ in $c_1$ and $c_2$
        $proportion \leftarrow \frac{n}{total}$
        if $proportion$ ¿ $maxProportion$ then
            $maxProportion \leftarrow proportion$
            $bestRange \leftarrow t$
        end if
    end for
```

Databait 9

Databait 9 measures the proportion of all entries that hold a specific column value for a column, for two different time points.

- Template: *Raw statistic* percent of *subject* / *column descriptor* / *column value* in *year*, compared to *raw statistic* percent in *year*.
- Sample: *11* percent of *CS professors* / *specialized in* / *Algorithms & Complexity* in *2020*, compared to *18* percent in *1991*.

```
Input: $c \leftarrow$ column name
Output: earliestYear, the earliest year in the dataset, latestYear, the latest year in the dataset, bestStartProportion,
    the percentage of all entries that have bestEntry for $c$ and earliestYear for time, bestEndProportion, the percentage
    of all entries that have bestEntry for $c$ and latestYear for time, and bestEntry, a column value in $c$ that maximizes
    the difference between bestStartProportion and bestEndProportion
    $earliestYear \leftarrow$ earliest year in the dataset
    $latestYear \leftarrow$ latest year in the dataset
    $entries \leftarrow$ the set of all unique column values in $c$
    $bestEntry \leftarrow None$
```

```
$bestStartProportion \leftarrow 0$
$bestEndProportion \leftarrow 0$
for $e \in entries$ do
    $startProportion \leftarrow$ percentage of entries that have $e$ for $c$ and $earliestYear$ for the time column
    $endProportion \leftarrow$ percentage of entries that have $e$ for $c$ and $latestYear$ for the time column
    $diff \leftarrow |startProportion - endProportion|$
    if $diff$ ¿ $|bestStartProportion - bestEndProportion|$ then
        $bestStartProportion \leftarrow startProportion$
        $bestEndProportion \leftarrow endProportion$
        $bestEntry \leftarrow e$
    end if
end for
```

Databait 10

Databait 10 compares one column value that is represented by more entries now than before, with one that is represented by less.

- Template: *Less OR More / subject / column descriptor / first column value* in *year* than in *year*, while *Less OR More* did in *second column value*.
- Sample: *Less / CS professors / specialized in / High-Performance Computing* in *2020* than in *2001*, while *more* did in *Robotics*.

```
Input: $c \leftarrow$ column name
Output: $earliestYear$, the earliest year in the dataset, $latestYear$, the latest year in the dataset, $bestUpEntries$, column values in $c$ which have grown most in proportion from $earliestYear$ to $latestYear$, and $bestDownEntries$, which have shrunk most
$earliestYear \leftarrow$ earliest year in the dataset
$latestYear \leftarrow$ latest year in the dataset
$entries \leftarrow$ the set of all unique column values in $c$
$upEntries \leftarrow$ empty array
$downEntries \leftarrow$ empty array
for $e \in entries$ do
    $startProportion \leftarrow$ percentage of entries that have $e$ for $c$ and $earliestYear$ for the time column
```

```
        endProportion ← percentage of entries that have e for c and latestYear for the time column
        diff ← $startProportion - endProportion$
        if diff ¿ 0 then
            upEntries.insert((e, diff))
        else
            downEntries.insert((e, —diff—))
        end if
    end for
    upEntries ← sort upEntries by diff, decreasing
    downEntries ← sort downEntries by diff, decreasing
    bestUpEntries ← first n entries in upEntries
    bestDownEntries ← first n entries in downEntries
```

Preliminary Databaits Validation Study: Computer Science Professors and Video Game Sales

In the age of misinformation, we believe it is responsible to conduct a preliminary study to pilot the insights created by the Databaits API. This is to ensure people find them believable, trustworthy, and interesting. All insights presented to participants in this preliminary study are true based on the tabular data given to the Databaits API. No personally identifiable information was collected from participants.

Preliminary Databaits Validation Study: Method We first introduced insights created by the Databaits API to the public through a preliminary study, which presented participants with insights created from two different tabular datasets (computer science professors and video game sales), then measured their reactions. Each participant read 24 different insights and then answered six follow-up questions per Databait. These questions measured whether participants found each insight interesting, believable, surprising, easy to understand, and if they wanted to learn more. Below is a sample set of questions:

Sample Databait: Over the past 25 years, the total number of games that were published by Nintendo and were classified as Sports nearly quadrupled.

- How interesting is this information? [Likert scale, 1 - "Not at all" to 6 - "Very"]
- How believable is this information? [Likert scale, 1 - "Not at all" to 6 - "Very"]
- How surprising is this information? [Likert scale, 1 - "Not at all" to 6 - "Very"]

- Is this sentence easy to understand? [Likert scale, 1 - "Not at all" to 6 - "Very"]
- Do you want to learn more about this information? [1 - "No" or 2 - "Yes"]
- Please share any thoughts or feedback on this sentence. [Open-ended]

This survey used a Likert scale with an even number of choices (1 to 6) so participants had to choose explicitly negative (1 to 3) or positive (4 to 6) answers. In addition, participants could only answer whether they wanted to learn more with a "Yes" or a "No." This binary choice best reflects whether users would choose to learn more in the real world. If an insight from the Databaits API were posted on Twitter, followed by a link to the source dataset it was created from, someone would either click on the link to learn more or decide not to click.

This survey also measured if participants' prior interest in the topics covered by the two datasets would react differently than those who did not. In other words, it sought to measure whether prior interest affected user reactions. Before participants saw any insights generated by the Databaits API, they answered 16 Likert scale questions with an even number of choices (1 to 6), indicating how interested they were in each of the 16 subtopics that later appeared in each insight in the survey. For example, one of Databaits' generated insights in the survey mentioned subtopics such as video games, Nintendo, and the Sports genre. Here are example questions participants answered:

- How interested are you in video games? [Likert scale, 1 - "Not at all" to 6 - "Very"]
- How interested are you in Nintendo? [Likert scale, 1 - "Not at all" to 6 - "Very"]
- How interested are you in Sports video games? [Likert scale, 1 - "Not at all" to 6 - "Very"]

We designed the survey to determine if some Databait types are more effective than others. Each Databait type appeared twice using the two different tabular datasets. The first tabular dataset is Drafty's Computer Science professors. Each row contains a CS professor's name, their subfield of expertise, the university where they teach, the year they were hired at that university, and the universities where they received their Bachelor's and Doctorate degrees. The second dataset is a table from Kaggle detailing historical video game sales. Each row contains the name of the video game, its publisher, the platform for which it was designed, its genre, the year it was published, and its sales globally: in North America, Europe, and Japan.

Preliminary Databaits Validation Study: Data Overview In total, the survey received 57 responses from paid crowdworkers on Amazon Mechanical Turk and 33 responses from those on Prolific. All paid crowdworkers were paid $2.25 per task (around $9 per hour). The median completion time for paid crowdworkers on Amazon Mechanical Turk was 16 minutes and 22 seconds. The minimum qualifications for paid crowdworkers on Amazon Mechanical Turk were an acceptance rate of 99% or above and the completion of at least 500 tasks. The median completion time for paid crowdworkers on Prolific was 17 minutes and 32 seconds. At the time we conducted this preliminary study, Prolific did not allow us to restrict availability by number of tasks completed or prior success rate per crowdworker.

We recruited paid crowdworkers from two different sources because the initial responses from Amazon Mechanical Turk were lower quality than we initially expected. Some Amazon Mechanical Turk responses were most likely written by bots, as evidenced by nonsensical answers to open-ended questions or no variance in Like scale answers (e.g., providing the same answer every time). After excluding participants who displayed these behaviors, there were 41 responses from Amazon Mechanical Turk and 32 from Prolific. These 73 responses are used in the following Results and Discussion section. The Results and Discussion section found no difference in the responses from Amazon Mechanical Turk and Prolific used in the final analysis.

Preliminary Databaits Validation Study: Results & Discussion Results from the preliminary study indicate that insights generated by the Databaits API are interesting, believable, and could be effective in attracting attention online and motivating people to engage with the source datasets. Participants indicated a positive response to insights (answered 4 or above on the 6-point scale) on Likert scale questions asking if they found the insight interesting 77% of the time, believable 91% of the time, and easy to understand 93% of the time. Participants only found the insight surprising 45% of the time. Participants indicated they did not want to learn more about the insight around 70% of the time. While this response might be less enthusiastic compared to the other questions, it still represents a high level of user interest compared to the attention most links receive on Twitter, where only 60% of links on Twitter receive a single click [75].

If the Databaits API could create tweets that were believable and lead people to learn more about the source, this could motivate integrating the Databaits API into Drafty to attract more visitors. Pearson's Correlation shows medium to large effects between a participant wanting to learn more about the insight and feeling it is interesting (r = 0.71, $p < 0.05$), believable (r = 0.46, $p < 0.05$), and easy to understand (r = 0.57, $p < 0.05$). These results indicate the insight alone might drive people to visit the source data. Pearson's Correlation shows no effect between a participant feeling surprised by the insight and wanting to learn more (r = -0.09). Databait's insights do not intend to motivate engagement or clicks by surprising people. Prior research has shown a correlation between clicks and people feeling surprised by misinformation campaigns [193].

Previous research shows relationships between someone's prior interest in a topic and increased engagement [51, 159]. This survey measures prior interest by participants answering 6-point Likert scale questions on their interest in the specific 16 subtopics (i.e., column values) featured across the insights:

1. colleges and universities
2. video games
3. Harvard University

4. Carnegie Mellon University

5. Artificial Intelligence

6. Human-Computer Interaction

7. Massachusetts Institute of Technology (MIT)

8. Brown University

9. Princeton University

10. Tsinghua University

11. Nintendo

12. Sony Computer Entertainment

13. Sports video games

14. Action video games

15. Shooter video games

16. Simulation video games

The answers to these Likert scale questions can be compared against how participants felt about individual insights they answered questions about later on in the survey. Pearson's Correlation shows prior interest in the column values was strongly correlated with whether the reader found the insight interesting ($r = 0.80$, $p < 0.05$). Prior interest also had moderate correlations with whether participants found the insights believable ($r = 0.67$, $p < 0.05$) and if they wanted to learn more ($r = 0.42$, $p < 0.05$). Pearson's Correlation shows a small effect between someone's prior interest and surprise had a slightly negative correlation ($r = 0.20$, $p < 0.05$), indicating someone's prior knowledge about specific information makes insights about that information less surprising. The insight being "easy to understand" showed a medium effect ($r = 0.42$, $p < 0.05$) with prior interest. This might be because the insights with simple structures remained straightforward regardless of what the insight was about. The correlation between someone's prior interest and their wanting to know more about the insight should help attract visitors to the system that stores the insight's source data. For example, Drafty is a system that hosts a source.

We also examined whether some Databait types are more effective than others. If one Databait type was found not believable or interesting, it should not be used to create and share insights in the wild. To determine this we used the interquartile range method (IQR) to compare the average responses to questions across all Databait types. First, it appears users prefer types with simple sentence structures over more complex ones (i.e., lower word counts and less punctuation). For example, Type 1 had high averages across all participants for whether participants found the insight interesting, believable, easy to understand, and wanting to know more. Type 1 is the most simple in structure and features only one column and one column value. It had the highest averages among all 10 types for being easy to understand (5.2) and making users want to learn more (1.4). Moreover, it had the second highest averages overall for being interesting (4.0) and believable (4.6). On the other hand, Type 4 is the most complex, involving two columns and three column values. It had the lowest average score for being interesting (3.6) and believable (4.1), along with the second lowest score for being easy to understand (4.6). In summary, insights generated by the Databaits API with simpler sentence structures were better at potentially attracting users. Along with Type 4, Type 10 is another complex Databait type that also showed relatively weaker results. It had the lowest average score for being easy to understand (4.6), as well as second lowest for being interesting (3.6) and having users want to learn more (1.2). However, these scores from the responses still fall within 1.5 times their respective interquartile ranges (IQR) from the first quartiles. Therefore, we decided that the relative weaknesses of Types 4 and 10 did not outweigh the diversity in sentence templates they provided and chose not to eliminate them from integrating the Databaits API into Drafty to create the new "Did You Know" feature.

"Did You Know": Providing Users Interactive Insights from the Databaits API within Drafty

The **Databaits API** creates automatic human-consumable insights derived from large tabular datasets, like Drafty's. In 2022, we developed the "Did You Know" feature in Drafty to bring these insights closer to Drafty's everyday visitors (i.e., unpaid contributors) by integrating the Databaits API within Drafty's spreadsheet interface. Now Drafty's everyday users can freely interact with insights generated from Drafty's real-time tabular data. "Did You Know" aims to provide something additional for people to interact with while using Drafty's spreadsheet interface. Hopefully, people will find the Databaits API's insights interesting and engaging and learn some insights. Prior research, SuggestBot [53], and Botivist [200], used sentence templates to automatically advertise a public system and attract people to contribute their time to specific causes freely. Botivist from Savage et al. [200] tweeted stories about Latin America's corruption issues. To build on these ideas, Drafty and its Twitter account present insights created using the Databaits API to current and potential visitors as a "Did You Know".

In Drafty, anyone can create an insight from the Databaits API using one of five methods:

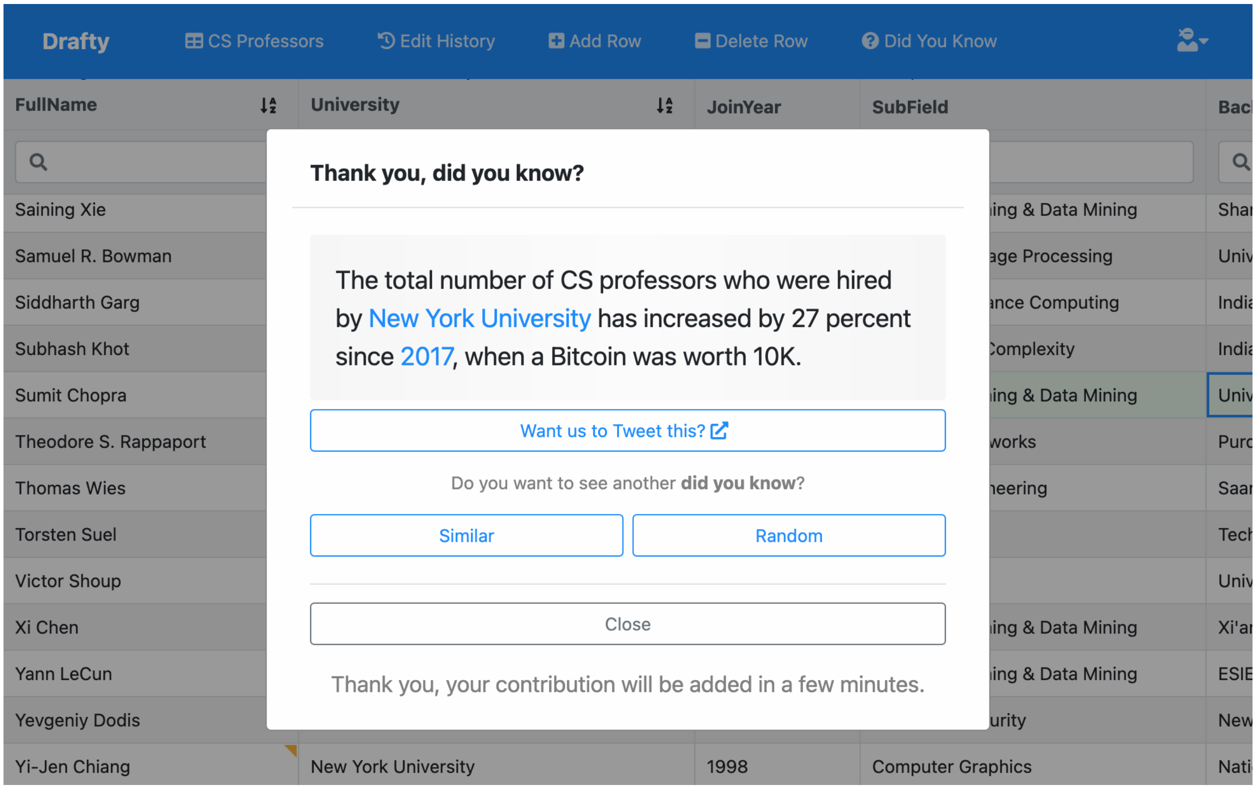
1. Right-click on a cell within the spreadsheet and select "See a Did you know" to generate a random insight using the selected row's information. This method lets users generate a "Did You Know" related to specific rows.
2. Select "Did you know?" in the top menu bar. This generates a random insight by selecting random columns and values from the tabular dataset. A user might not have a cell selected, so this method allows them to generate a random insight without having first to edit data.

3. Every time a user edits data (changing a cell's value, adding a new row, or deleting a row), the system generates a random "Did You Know" based on that row's information. This gives the user a small reward for making an edit and another method to help them discover and interact with a "Did You Know."

4. When viewing the Did You Know Modal, a user can choose to see another "Similar." This generates a random type of insight using the columns and input values used to generate the previous insight. For example, if the previous insight used the values "Databases" and "McGill University", the next insight would create an insight using at least one of these values.

5. When viewing the Did You Know Modal, a user can choose to see a "Random." This generates a random insight using the same method as Menu-Item-Random. This method allows users to continuously see a random "Did You Know" created from all of Drafty's tabular data.

Drafty's users can freely generate, see, and interact with as many insights as they want. Drafty will also show users an insight using the "Did You Know" interface post-contribution as part of a formal *"thank you"* and confirmation message. Previous research shows gratitude can increase unpaid contributions [164, 237]. These methods allow for analysis to study if users' next actions are beneficial to the health of the data. For example, do users make accurate contributions, stay engaged by searching for more data, or create and read more insights?



FullName	University	JoinYear	SubField	Bac
Saining Xie			ing & Data Mining	Shar
Samuel R. Bowman			age Processing	Univ
Siddharth Garg			ance Computing	India
Subhash Khot			Complexity	India
Sumit Chopra			ing & Data Mining	Univ
Theodore S. Rappaport			works	Purc
Thomas Wies			neering	Saar
Torsten Suel				Tech
Victor Shoup				Univ
Xi Chen			ing & Data Mining	Xi'ar
Yann LeCun			ing & Data Mining	ESIE
Yevgeniy Dodis			urity	New
Yi-Jen Chiang	New York University	1998	Computer Graphics	Nati

Thank you, did you know?

The total number of CS professors who were hired by New York University has increased by 27 percent since 2017, when a Bitcoin was worth 10K.

Do you want to see another **did you know**?

Thank you, your contribution will be added in a few minutes.

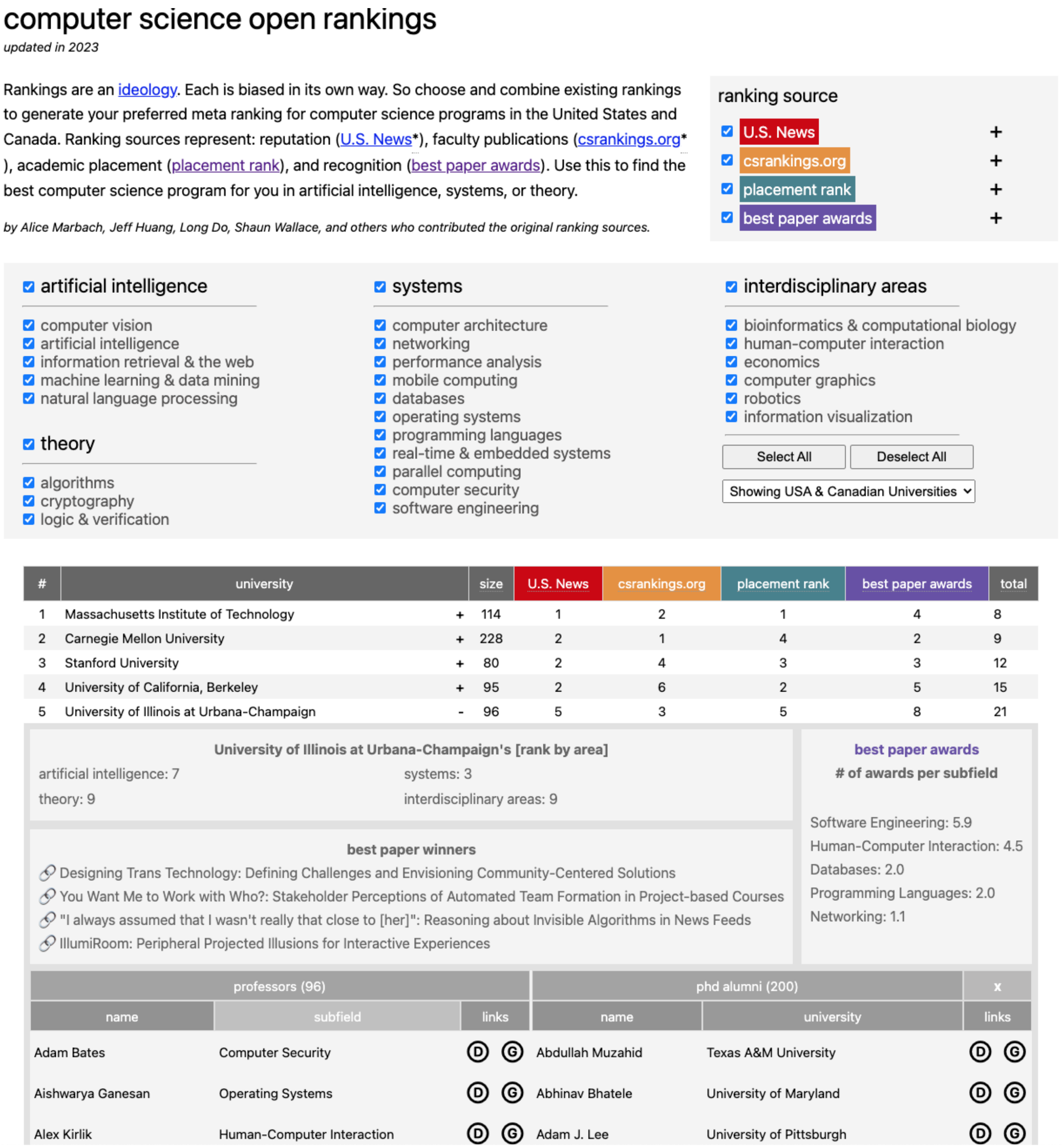
Figure 7.3: The Did You Know Modal is presented after someone intentionally creates a Did You Know or edits data in Drafty. If the visitor created an insight within Drafty, this interface would remove the wording relating to the confirmation thank you message. Within the modal, visitors can perform several different interactions. They can choose for the system to tweet the "Did You Know" to our Twitter account. See another "Did You Know" generated using similar values from the current insight. See another "Did You Know" randomly generated from the tabular data. Users can select the blue text from the sentence to close the window and automatically search the spreadsheet for rows matching the selected value. Close the window and return to the spreadsheet interface by selecting the "X" in the top right corner, pressing the ESC key, or selecting the button labeled "Go back to editing."

Computer Science Open Rankings: a Dynamically Updated Meta-Ranker from Drafty's Computer Science Professor Profiles

Computer Science Open Rankings is an approach to computer science rankings developed by Brown University HCI Lab members. Rankings are an ideology, and each can be biased. It is a meta ranker using real-time data from CS Professors (Drafty) as one of its sources. Its rankings and data are updated dynamically every five minutes. In CS Open Rankings, users freely choose and combine existing rankings to generate their preferred meta-ranking for computer science programs in the United States and Canada. The four ranking sources are (more detailed descriptions are below):

1. reputation (U.S. News)
2. faculty publications (counting papers published from top conferences)
3. academic placement (placement rank)
4. recognition (best paper awards)

People can use this resource to find the best computer science program in artificial intelligence, systems, or theory. Or, they can filter the results further by subfields, such as Human-Computer Interaction or Databases, see Figure 7.4. In CS Open Rankings, each university lists its current tenure-track Computer Science faculty and alums who are tenure-track Computer Science faculty at another university listed. Each faculty listing has a link to Drafty to view their full academic profile.



computer science open rankings

updated in 2023

Rankings are an ideology. Each is biased in its own way. So choose and combine existing rankings to generate your preferred meta ranking for computer science programs in the United States and Canada. Ranking sources represent: reputation (U.S. News*), faculty publications (csrankings.org*), academic placement (placement rank), and recognition (best paper awards). Use this to find the best computer science program for you in artificial intelligence, systems, or theory.

by Alice Marbach, Jeff Huang, Long Do, Shaun Wallace, and others who contributed the original ranking sources.

ranking source

☑ U.S. News +
☑ csrankings.org +
☑ placement rank +
☑ best paper awards +

☑ artificial intelligence

☑ computer vision
☑ artificial intelligence
☑ information retrieval & the web
☑ machine learning & data mining
☑ natural language processing

☑ theory

☑ algorithms
☑ cryptography
☑ logic & verification

☑ systems

☑ computer architecture
☑ networking
☑ performance analysis
☑ mobile computing
☑ databases
☑ operating systems
☑ programming languages
☑ real-time & embedded systems
☑ parallel computing
☑ computer security
☑ software engineering

☑ interdisciplinary areas

☑ bioinformatics & computational biology
☑ human-computer interaction
☑ economics
☑ computer graphics
☑ robotics
☑ information visualization

#	university		size	U.S. News	csrankings.org	placement rank	best paper awards	total
1	Massachusetts Institute of Technology	+	114	1	2	1	4	8
2	Carnegie Mellon University	+	228	2	1	4	2	9
3	Stanford University	+	80	2	4	3	3	12
4	University of California, Berkeley	+	95	2	6	2	5	15
5	University of Illinois at Urbana-Champaign	-	96	5	3	5	8	21

University of Illinois at Urbana-Champaign's [rank by area]

artificial intelligence: 7 systems: 3
theory: 9 interdisciplinary areas: 9

best paper winners

- Designing Trans Technology: Defining Challenges and Envisioning Community-Centered Solutions
- You Want Me to Work with Who?: Stakeholder Perceptions of Automated Team Formation in Project-based Courses
- "I always assumed that I wasn't really that close to [her]": Reasoning about Invisible Algorithms in News Feeds
- IllumiRoom: Peripheral Projected Illusions for Interactive Experiences

best paper awards
# of awards per subfield

Software Engineering: 5.9
Human-Computer Interaction: 4.5
Databases: 2.0
Programming Languages: 2.0
Networking: 1.1

professors (96)			phd alumni (200)		x
name	subfield	links	name	university	links
Adam Bates	Computer Security	Ⓓ Ⓖ	Abdullah Muzahid	Texas A&M University	Ⓓ Ⓖ
Aishwarya Ganesan	Operating Systems	Ⓓ Ⓖ	Abhinav Bhatele	University of Maryland	Ⓓ Ⓖ
Alex Kirlik	Human-Computer Interaction	Ⓓ Ⓖ	Adam J. Lee	University of Pittsburgh	Ⓓ Ⓖ

Figure 7.4: CS Open Rankings is a dynamically regenerated web page that uses Drafty's evolving dataset of Computer Science professor profiles as its source data. Users can freely visit and interact with the system. They can select the "D" icon to visit Drafty and automatically filter its spreadsheet interface to show that professor's information.

Reputation (U.S. News) is a ranking based on the reputation of graduate programs, compiled from the survey responses of academics to produce a peer assessment score per university.

Faculty publications measures research publications by professors in computer science. Publications occurring in selected conferences in the year 2010 or later are counted based on the proportion of authors that are professors at that institution. The numbers CS Open Rankings report under the ranking might differ from csrankings.org [17] because while it uses the same formula (geometric mean count), it has a different implementation and uses different subfields and source data for the professors.

Placement Rank is based on the academic placement of students from an institution into computer science faculty positions in the United States and Canada. The data comes from Drafty's publicly-editable dataset of professor profiles. Placement Rank uses the PageRank algorithm and is inspired by the PageRank analysis by digital historian Ben Schmidt [201]. The university where a professor received their Bachelors or Doctorate is the source node, and their currently affiliated university is the destination node in the PageRank algorithm. A university endorses the degree-granting institutions of the person they are hiring, as those institutions either trained or attracted a high-quality student. It is relatively difficult to manipulate this metric because the best strategy for a degree program is to convince other universities to hire your alums.

Recognition (best paper awards) measures papers from 30 top computer science conferences recognized as "best papers". Points are assigned to schools based on author affiliation and their position in the author list based on a decreasing exponential scale. The list of paper awards has been maintained by Professor Jeff Huang from the Brown University Computer Science Department and other members of the Brown University HCI Lab [3]. Jeff Huang is one of the authors of this paper.

7.4 Naturalistic Study: Attract and Engage Everyday Users In The Wild

This research features two approaches for evaluation: descriptive and intervention. This approach focuses on evaluating the natural usage of the Drafty system after introducing new features, notably Databaits and CS Open Rankings. Simultaneously, we will share several websites and resources containing static insights from Drafty's data. Introducing these features and resources serves as an intervention to Drafty's naturally occurring user base and community.

The evaluation will study three dependent variables: the accuracy of edits per visit, the number of edits per visit, and the average user engagement per visit. These measures are captured anonymously by the Drafty web application. An increase in the dependent variables would indicate that Drafty's community is helping to edit, review, and maintain its public data's accuracy. The evaluations for the descriptive and intervention approaches will analyze these dependent variables. However, this evaluation of naturalistic usage has several confounding variables, such as seasonality, whether

[3] https://jeffhuang.com/best_paper_awards/

it is someone’s first visit or how they arrived at Drafty.

7.4.1 Naturalistic Study: Recruitment Methods & Study Design Considerations

We reviewed the following methods, procedures, and proposed analysis with our local Institutional Review Board. They considered the systems, methods, and analysis to be user experience research because of the naturalistic study methods and because all of Drafty’s users are anonymous. To enable a naturalistic study, we developed a completely new version of Drafty to anonymously track user interactions and allow the system to be better search indexed and discovered by organic visitors.

7.4.2 Methods to Recruit (Source) Visitors to Drafty: Organic, Dynamic, Static, and Asking

Users freely came to Drafty from one of four sources: organic, dynamic, static, and asking. This section explains how each source was developed and deployed in the wild to attract and engage users. The back-end web services can track users from specific URLs, but they do not track their IP address or record any personally identifiable information. Most links exist at publicly available web applications such as CS Open Rankings, Twitter, or forums such as Reddit.

Organic: Everyday Visitors Discovering and Visiting Drafty

“Organic” visitors can freely visit the Drafty web app through various means, such as search engines, typing the URL in the browser, or other people sharing a link to Drafty’s public web app. We developed the PolyMu client-side javascript library to make Drafty’s thousands of rows of Computer Science professor profiles easier to search index by search engines such as Google. Chapter 5 studying publicly available tabular datasets shows they can rely on everyday visitors typing in the URL or coming from search engines to become unpaid contributors.

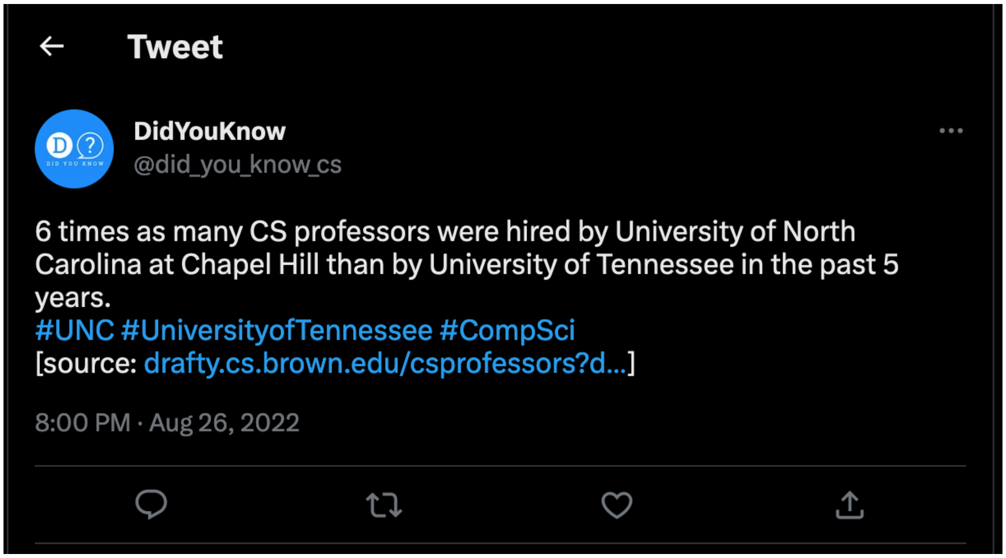
Dynamic: Computer Science Open Rankings and Tweets Containing Insights from the Databaits API

“Dynamic” visitors can arrive at Drafty by either selecting links on CS Open Rankings or tweets from Drafty’s Twitter account. The “dynamic” indicates the information presented in CS Open Rankings and Twitter is dynamically created or regenerated from Drafty’s tabular dataset. They evolve as Drafty’s data changes. The core idea is people would visit this public information and then select a link to view and possibly edit the source data on Drafty.

CS Open Rankings provides a link back to Drafty under every professor or alumni it lists per university. Drafty will automatically search its spreadsheet to display that professor’s full academic profile to the end user.

Another way “dynamic” visitors can arrive at Drafty is by selecting a link from a tweet on Drafty’s public Twitter account; see Figure 7.5 for an example. These tweets are generated in two different ways. They are generated once daily by Drafty itself. Also, visitors within the Did You Know Modal (see Figure 7.3) visitors can choose to have

Drafty tweet the insight to Drafty's Twitter account. Each Databait is tweeted using a consistent template that supports all Databait types. See Figure 7.5 for examples. These templates do not use language considered nudging based on prior recommendations from Savage et al. [200]. The goal is to be transparent and communicate the data source that generated the insight to potential visitors.



DidYouKnow
@did_you_know_cs

6 times as many CS professors were hired by University of North Carolina at Chapel Hill than by University of Tennessee in the past 5 years.
#UNC #UniversityofTennessee #CompSci
[source: drafty.cs.brown.edu/csprofessors?d...]

8:00 PM · Aug 26, 2022

Figure 7.5: Drafty's public Twitter account (@did_you_know_cs) "DidYouKnow." Each tweet contains an insight generated using the Databaits API. Users can select the link to go to Drafty's Computer Science Professor tabular dataset from each tweet.

Each tweet contains a link to the Drafty Computer Science professor dataset. Thus, users on Twitter can visit and edit the source data by selecting the URL embedded in the tweet. This embedded URL contains attributes that allow Drafty to track the tweet containing the Databaits visitors selected. This helps to evaluate how these tweets can help attract visitors to Drafty's public dataset.

Static: Creating CS Open Data, a Static Website with Analysis and Insights derived from Drafty's Computer Science Professor Profiles

CS Open Data is one of the "static" sourcing methods that consists of four separate web pages that contain a written static analysis of Drafty's Computer Science professor dataset:

1. CS Faculty Composition and Hiring Trends
2. Bias in Computer Science Rankings
3. Who Wins CS Best Paper Awards?
4. Verified Computer Science Ph.D. Stipends

These static web-based articles can answer questions like “what percentage of professors get their bachelor’s and PhD from the same institution?”, “which institutions hoard most of the best paper awards?”, “what stipend do ivy league universities pay?”, “how is U.S. news biased in ranking computer science programs?”, “how many machine learning professors do a specific university have?”, or “what computer science areas have been growing?”. These are the type of questions from users of Drafty we have seen over multiple years. We believe these static analyses can answer these questions and motivates people to view and edit the raw data on Drafty that created them.

Static: Posting on Forums to Inform People about Drafty

The second of the “static” sourcing methods consists of posting about Drafty’s Computer Science professor dataset on public forums like Reddit. This sourcing method to recruit visitors is similar to prior research efforts [225]. An example title used on Reddit was “Some Interesting Insights on CS Depts and Faculty in the US/Canada” and an example post sharing Drafty contained the text:

> “Hi, I just wanted to share a resource for the CS community we have developed at the Brown HCI lab. We have helped upkeep a dataset of tenure-track CS faculty in the US/Canada for 8 years.
> The system (and users) generate automatic insights from the data, for example: Over the past 10 years, the total number of CS professors who were hired by Stanford University and specialized in Computer Graphics more than doubled.”

These static forum posts shared an insight generated using Databaits from Drafty. This decision ensured these posts shared a static insight like CS Open Data and advertised the new automatic insights feature within Drafty. This sourcing method takes little initial effort and should be easy to replicate for others.

Asking: Providing a Personal Touch by Emailing and Tweeting People to Visit, Review, and Contribute to Drafty’s Computer Science Professor Dataset

Building on ideas from Savage et al. [200] and Brady et al. [28], that shows the best method to elicit contributions is to ask people or your friends, I (Shaun Wallace) to ask people to visit and contribute to Drafty’s data two different ways. The first was to tweet replies to people who recently announced they were hired to be a Computer Science tenure-track faculty member at a university listed in Drafty, for example:

> “Congrats ¡first_name¿! We have been building a public dataset of computer science professors to help students to find potential advisors. You should add/update yourself as a prof! :)”

I publicly replied to 25 people on Twitter over the Spring and Summer of 2022. Tweeting others takes continued effort but should be simple to replicate. These tweets act as a call to action for someone from the community to contribute their data. The tweets directly link to Drafty. Because they are posted publicly, anyone can select the link to visit Drafty anonymously. Therefore, any contributions from the link could come from anyone.

The second way Shaun Wallace asked for contributions was to email the department chairs and directors of Computer Science departments listed in Drafty using his Brown University email address, for example:

> “ [number_of_faculty] faculty from [university_name] are listed in this public dataset of CS professors’ data, a resource we’ve been compiling for the CS community.
>
> CS Open Rankings uses this data, which features your department.
>
> According to CS Open Rankings, [university_name] is ranked ¡ranking_number¿ in [ranking_type] among US and Canadian Universities.
>
> The listings consist of tenure-track faculty members who can advise CS PhD students. If there are any corrections for your current faculty or alumni, please feel free to edit the CS professors’ data. We’re also happy to share the raw data if you contribute. ”

The text in the email, “CS professors’ data” links directly to Drafty and has a call to action for a department to review their faculty and alumni’s data. Since I emailed multiple people each time, anyone could have forwarded the email to others in their department or chosen to contribute to Drafty.

7.5 Naturalistic Study: Results

A public data system relies on waiting for users to edit and improve a dataset’s accuracy over time. Previous research shows that engaged users who visit multiple times and interact more with a system will potentially make more accurate contributions. This analysis is conducted per visit because Drafty tracks anonymous visitors over time. This analysis assesses how features developed to improve a public data system can increase edits per visit, accuracy per visit, and user engagement per visit. The analysis for this naturalistic study splits users into different groups based on confounding factors:

1. Visits in 2022 versus 2021 from March 16th through September 28th.

2. Was the visit a user’s first visit or not?

3. Visits from users who only made accurate edits versus those who did not.

4. Which of the four resources did the user come from? (organic, static, dynamic, or were they asked)

5. Visits where users freely chose to create at least one Databait.

This analysis observes and discusses the dependent variables per group to understand the editing and engagement behaviors of Drafty's everyday users within a real-world scenario covering multiple months.

To test if each group's data is normally distributed, the analysis uses the D'Agostino-Pearson test when the sample size is greater than 5,000. Otherwise, the Shapiro-Wilk test is used. Unless otherwise stated, all reported significant results from multiple comparisons to pass the Holm-Bonferroni correction [1, 207].

7.5.1 Naturalistic Study: Data Overview

The analysis for the naturalistic study uses data collected over 196 days from March 15th through September 28th of 2022. Each date was chosen because of the introduction of new features within Drafty. On March 15th of 2022, we introduced the "Did You Know" feature utilizing Databaits. September 28th marks the beginning of recruiting users within Drafty to participate in a pilot survey for a new study.

All users included in this analysis made at least one interaction within Drafty (i.e., click, search, edit, etc.). This inclusion criteria ensures each user in the analysis is not a bot and should benefit the interpretation of the results. After applying this inclusion criteria, in 2022, there were 57,700 total visits, 290,416 interactions, and 3,438 edits. During the same time period in 2021, there were 2,536 total visits, 86,950 interactions, and 265 edits. Comparing visitors from 2022 versus 2021, in 2022, there were 22.8 times more visits, 3.3 times more interactions, 6.8 times more average interactions per visit, and 13 times more edits per visit compared to visitors in 2021 during the same range of months.

To compute accuracy per visit, we used stratified sampling to select edits to label as correct or incorrect. We used faculty web pages, CVs, and LinkedIn profiles to check if the edit was correct when it was made. We reviewed and labeled 1,709 edits from 227 visits for 2022 and 265 edits from 55 visits for 2021. The average accuracy per visit in 2022 was 95%. While users in 2021 were 73% accurate per visit. When computing average accuracy per visit, we only used visits with at least one edit we checked by hand. The following sections will evaluate these results in greater detail.

7.5.2 Naturalistic Study: Metrics

This section covers the metrics used to analyze Drafty's data. All metrics described below are computed per visit. The back-end server uses cookies to identify web browsers that access Drafty. A visit is defined as a unique individual with successive interactions within 20 minutes of each other. This can account for natural short breaks within a single visit. For example, to answer a question from a family member or use the bathroom.

- *Edits:* The number of contributed data points (i.e., edits) per visit.

- *Conversion Rate:* Did the visit make an edit?

- *Accuracy:* The number of contributed data points marked as correct over the total number of contributions checked per visit. For example, a new row contributes six data points, while editing a single cell is one data point.
- *Interactions:* Within a visit, the total number of recorded interactions within Drafty. See Table 7.1 for a list of the interactions Drafty records.
- *Session Length:* Within a visit, the number of seconds between the first and last recorded interaction with Drafty.
- *Engagement Score:* Within a visit, the total number of recorded interactions multiplied by their weights (See Table 7.1 for interactions and their weights.) plus the session length's weight (short = 1 point, medium = 4 points, long = 8 points) within Drafty. The 25th and 75th quartiles determine the weight of the session length during the 2022 study period. Short session lengths are below the 25th quartile. Medium session lengths are between the 25th and 75th quartiles. Long session lengths are above the 75th quartile. Session lengths directly indicate how long a visitor is engaged with Drafty during their visit. Contributions take the most effort and benefit Drafty's long-term goal of maintaining evolving data. Thus, they are provided the highest weight. The idea to assign interaction weights comes directly from building the User Interest Profile in Chapter 3.

7.5.3 Visitors in 2022 are more Accurate and Engaged compared to 2021

This section compares the dependent variables per visit from 2022 to 2021 to account for seasonal factors affecting users' motivations and contribution behaviors.

Accuracy per Visit

The average accuracy per visit in 2022 (M = 95%, SD = 0.20, N = 227) compared to 2021 (M = 73%, SD = 0.42, N = 55) is around 22-points higher. A Mann-Whitney U test reveals this 30.5% increase in accuracy per visit in 2022 is statistically significant, $U = 7928$, $p < 0.001$. This indicates the new features implemented in Drafty in 2022 helped to increase the accuracy of edits per visit.

Prior research using similar data reports accuracy per edit. In this chapter, accuracy per edit in 2021 is 92%, and in 2022 it was 98%. The average accuracy per edit in 2022 (M = 98%) is higher than similar research from 2016–2017 (M = 76%) [225], and 2017–2019 (M = 89%) [224]. While those two studies were able to create edits with better accuracy when nudging users to edit data matching their interests, in 2016–2017 (M = 88%) [225], and 2016–2019 (M = 95%) [224]; our study's visitors achieved similar or better accuracy without the system asking visitors to contribute. With Drafty's new features in 2022, visitors were more accurate than previous work without employing de-motivating tactics such as nudging (i.e., nudging), which prior research shows decreases contributions over time [200].

Classification	Interaction Type	Weight
Modify Tabular Data	edit cell	4
	add new row	24
	delete a row	4
	visit from a source	1
Interact with Cells	click a cell	1
	double-click a cell	1
	select multiple cells	1
	copy a cell	1
	paste value into a cell	1
Interact with Columns	sort by column	1
	search single column	1
	completed single column search	1
	completed multiple column search	1
	search multiple columns	1
	clear a column's search	1
	copy all values from column	1
Interact with Notes per Row	add a note to a row	4
	vote thumbs up on a row's note	2
	vote thumbs down a row's note	2
	deselect your vote on a row's note	2
	open the note's modal window to read notes	1
Create an insight	right-click a row and select create an insight	2
	create a similar insight within the Did You Know Modal	2
	create a random Databait within the Did You Know Modal	2
	tweet an insight to Drafty's Twitter account	2
	select a value from an insight to auto-search Drafty	1
Other Interactions	right-click a row and search Google for that row's data	1
	view a different web page on Drafty	1

Table 7.1: This table shows the recorded user interactions with Drafty and their weights (T_w) used to build the engagement score per visit. Adding a new row is 24 points total because there are six columns in Drafty, and one contribution (edit cell) is allotted 4 points. When users add a new row, they contribute six data points per row. Deleting a row is 4 points because the user takes a single action to delete a row.

Edits per Visit

This analysis compares the number of visits in 2022 versus 2021 who made at least one interaction. In 2022, 406 visitors made at least one edit, and 6,577 did not, for a conversion rate of 5.8%. In 2021, 63 visitors made at least one edit, and 2,473 did not, for a conversion rate of 2.5%. A chi-squared test reveals these frequencies were significantly different, $X^2(1, N = 9519) = 43.3$, $p < 0.001$. Turning visitors who do not contribute (i.e., lurkers) into contributors is a complex scenario within crowd-powered and peer production systems such as Drafty [18]. Thus, Drafty's new features deployed in 2022 helped to attract visitors who were twice as likely to make at least one edit compared to 2021. Building on this result, visits in 2022 made 3.7 more edits per visit than in 2021. A Mann-Whitney U test reveals this increase in edits per visit in 2022 (M = 0.49 SD = 4.91, N = 6983.00) compared to 2021 (M = 0.13 SD = 2.00, N = 2536) is statistically significant, $U = 9151683$, $p < 0.001$. In 2022, visitors made more frequent edits than in 2021, thus providing enough contributions to maintain the accuracy of Drafty's data.

In Chapter 5, we report everyday users editing Computer Science professor data made 2.6 edits daily over multiple years. In our study, visitors in 2022 made 17.5 edits per day. Thus, comparing these similar efforts, our visitors made 6.7 more edits per day compared to those in Chapter 5. Although visitors in 2021 only made 1.4 edits per day, Chapter 5 explains why the edits per day in 2021 were lower than our prior research [224]. We reported the lowest months for edits were May and June, included in our study data, while the months with the highest edits were December and January, which are not included in our data. These observations show how the increased edits per day in 2022 will benefit the overall accuracy of the data, given how the seasonal confounds may de-motivate the average visitor.

Engagement per Visit

In crowd-powered and peer production systems, creating an engaged user base over time is essential to maintain accurate data [16]. We will compare the average interactions, visit length in seconds, and user engagement profile per visit from visitors in 2022 and 2021. The average number of interactions per visit was similar in 2022 (M = 34 SD = 68, N = 6983) and 2021 (M = 34 SD = 93, N = 2536). While we found significance in comparing visits from 2022 and 2021, the effect size was 0. Because Drafty does not record scrolling behaviors as individual interactions, session length is another indicator of engagement per visit. The average seconds per visit in 2022 (M = 340 SD = 840, N = 6983) compared to 2021 (M = 232 SD = 580, N = 2536) was 1 minute and 48 seconds longer. A Mann-Whitney U test reveals this 47% increase in session length is statistically significant, $U = 10485543$, $p < 0.001$. We can use the engagement score to understand if visitors were engaging in interactions with Drafty that require more effort than merely clicking on cells while lurking. The average engagement score in 2022 (M = 42 SD = 73, N = 6983) compared to 2021 (M = 39 SD = 95, N = 2536) was around 3 points higher. A Mann-Whitney U test reveals this 6% increase in the engagement score is statistically significant, $U = 10056979$, $p < 0.001$. Overall, results show visitors were more engaged with Drafty in 2022 compared to 2021. The new features deployed in 2022 achieve this through mostly automated methods compared to more traditional methods from paid crowdsourcing that require manual engagement with users [178]. This makes it feasible for someone managing a public data system, like Drafty, to focus their time on building a better dynamic system for the users rather than managing the users themselves.

7.5.4 Highly Accurate Visits are more Engaged than Lurkers

This section analyses the relationships between accuracy per visit, edits per visit, editing behaviors, and engagement between accurate visits (i.e., visits that made only accurate edits) and visits with no edits (i.e., lurkers). This analysis will help us to understand what behaviors indicate and explain highly accurate visits compared to other visits.

Accuracy per Visit

Among visits where we labeled at last one edit as correct or incorrect, 92% of visits made only accurate edits. In contrast, only 8% (19 total visits) made at least one error. The sample size for total visits that made at least one error is too small to conduct hypothesis testing. Therefore, we will study other behavior patterns among accurate visits and common errors when editing Drafty's data. The common mistakes made during visits that made at least one incorrect edit were:

1. Accidentally selecting the first value from the modal edit drop-down. There were cases when visitors would make this error but immediately corrected it. These mistakes were not included in the analysis.

2. Sharing information that was not yet public or true. For example, someone leaving a university.

3. Entering a research area that was a professor's secondary research area, not their primary.

4. The most common error was Bachelor's degree. There was no obvious pattern to errors. The overall accuracy for all visitors' edits to Bachelor's degrees in 2022 was 96%.

One of the most interesting observations is among visitors who immediately corrected their mistakes. The visitors who accidentally selected the first value from the modal edit drop-down and immediately corrected their mistake never made any errors.

Edits per Visit

In 2022, many of Drafty's visitors made highly accurate contributions. The 227 total visits in 2022 that we checked by hand made 12.5 edits per visit with an accuracy of 95% per visit. Pearson's Correlation reveals no effect between accuracy and the number of edits per visit (r = 0.08, p = 0.25). This result replicates our previous research showing no relationship between the number of edits and accuracy from Chapter 5. The lack of a relationship between accuracy and the number of edits is contrary to results from Wikipedia, where a few users contribute the bulk of accurate edits for its unstructured data [96, 132]. While Drafty has thousands of visitors, it does not attract the same number of visitors as Wikipedia. These results would likely shift if Drafty's users were vandalizing data (i.e., making intentional and frequent errors).

Engagement per Visit

This analysis compares accurate visits (i.e., visits that made only accurate edits) and visits with no edits. Visits with no edits (i.e., lurkers) made at least one interaction to ensure they were real people. The average number of interactions per visit for accurate visits (M = 63 SD = 72, N = 208) compared to visits with no edits (M = 32 SD = 65, N = 6577) was

1 minute and 48 seconds longer. A Mann-Whitney U test reveals this 47% increase in session length is statistically significant, $U = 1009201$, $p < 0.001$. The average session length per visit for accurate visits (M = 707 SD = 72, N = 208) compared to visits with no edits (M = 324 SD = 65, N = 6577) was 6 minutes and 23 seconds longer. A Mann-Whitney U test reveals that the average session length for accurate visits being 2.2 times longer than visits with no edits is statistically significant, $U = 1040255$, $p < 0.001$. The average engagement score for accurate visits (M = 106 SD = 114, N = 208) compared to visits with no edits (M = 38 SD = 67, N = 6577) was around 68 points lower. A Mann-Whitney U test reveals that the engagement score for accurate visits being 2.8 times larger than visits with no edits is statistically significant, $U = 5701293$, $p = 0.03$.

Overall, we can see that visits that only made accurate edits were highly engaged compared to lurkers. Our analyses revealed that accurate visits had the highest engagement score. Among only accurate visits, Pearson's Correlation reveals a small effect between accuracy and engagement score per visit in 2022 (r = 0.18, p ¡ 0.001). The engagement score shows the potential to understand behaviors even among a cohort of visitors with similar interests in Computer Science professor data. While this provides some evidence of the benefits of running a public data system to maintain data, Chapter 8 reveals what factors motivate contributions within Drafty.

7.5.5 First-Time Visits are Less Engaged but Just as Accurate as Return Visits

This section compares the dependent variables per visit between first-time and return visits. This analysis will help us understand if first-time visitors can also make accurate contributions to a public data system like Drafty.

Accuracy per Visit

While there is a 4% increase in average accuracy per visit when comparing first-time visits (M = 0.94% SD = 0.24, N = 136) and return visits (M = 0.98% SD = 0.12, N = 91), a Mann-Whitney U test reveals this increase is not statistically significant. This result replicates our previous research on Drafty's data from Chapter 5, showing first-time visits and return visits make accurate edits. This result for that first-time visitors with mutual interests makes accurate edits to a public data system runs contrary to prior research on Wikipedia editors, where returning users make more accurate contributions [92, 132]. This finding replicates results from Chapter 5, showing that first-time visitors are just as accurate as return visitors. To maintain a dataset over time, a public data system like Drafty can recruit first-time and return visitors to make accurate contributions.

Edits per Visit

This analysis compares the frequency of first-time and returning visits that made at least one edit. There were 259 first-time visits that made at least one edit, and 4,303 did not. The conversion rate for first-time visits is 0.06%. In contrast, the conversion rate for return visits is also 0.06%. There were 147 return visits that made at least one edit, and 2,274 did not. A Chi-Square test reveals these frequencies were not significantly different. These similar conversion rates show that first-time visits are just as likely to contribute as returning visitors. Understanding which users will contribute is a long-standing research question [242]. Chapter 3 shows that return visitors are more likely to make contributions. While return visits (M = 0.72 SD = 6.73, N = 2421) make around 2 more edits per visit compared to first-time visits (M = 0.37 SD = 3.58, N = 4562), a Mann-Whitney U test reveals this increase is not statistically significant. These results show it might be more important to focus on recruiting and engaging all types of users rather than creating a small group of users who make the majority of contributions.

Engagement per Visit

When developing a public data system, should the focus be on motivating first-time visits or engaging people to make return visits? This analysis compares the average interactions, visit length in seconds, and user engagement profile per visit between first-time and returning visits in 2022. The average number of interactions per visit for first-time visits (M = 33 SD = 65, N = 4562) compared to return visits (M = 36 SD = 74, N = 2421) was around three interactions less per visit. A Mann-Whitney U test reveals this 9% decrease in interaction per visit is not significant. The average seconds per visit for first-time visits (M = 214 SD = 503, N = 4562) compared to return visits (M = 575 SD = 1213, N = 2421) was 6 minutes and 1 second shorter. A Mann-Whitney U test reveals this 63% decrease in the session length is statistically significant, $U = 5305198$, $p = 0.007$. The significant increase in session time for return visits could indicate they are exploring more of the data. Return visits' standard deviation for session length is more than double that of first-time visits, indicating a greater variation in session lengths among return visits. The average engagement score for first-time visits (M = 40 SD = 68, N = 4562) compared to return visits (M = 44 SD = 81, N = 2421) was around 4 points lower. A Mann-Whitney U test reveals this 9% decrease in the engagement score is statistically significant, $U = 5701293$, $p = 0.03$. This difference can mainly be explained by the longer session times of return visits. Unlike research on Wikipedia editors, which suggests are more robust onboarding experience to retain first-time visitors [171], these results show first-time visits make accurate edits while having shorter session times than return visits within Drafty. Overall, these results indicate return visitors might be more curious and want to explore the system and its data. Thus providing additional features to engage users within a system could be beneficial to sustain engagement.

7.5.6 Visitors from Dynamic Recruitment Methods that Use Automatic Insights and Asking People to Contribute are the most Accurate

Previous research shows that sending personal messages asking others to contribute to altruistic causes will motivate accurate contributions [160, 200]. This same research shows automated methods to nudge users to contribute work in the short-term but does not elicit the long-term contributions necessary to maintain a continued effort. Thus, this section's analysis compares the accuracy, the number of edits, and engagement based on how the visits were sourced (i.e., recruitment method): organic, dynamic, static, and asking.

	Visits	Edits	Edits per Visit	Accuracy per Visit	Min. Accuracy
organic	3408	1509	0.44	94.8%	0%
dynamic	1402	1220	0.87	99.1%	80%
static	1785	278	0.16	88.0%	0%
asking	387	429	1.11	99.6%	86%

Table 7.2: Visitors recruited through dynamic sources or by personally asking them were over 99% accurate. With the minimum accuracy per visit no lower than 80%. Dynamic sources almost garnered as many edits as organic visits but did so at a much higher accuracy. Making static analysis or forum posts yielded a lower number of edits and average accuracy.

Accuracy per Visit

While the accuracy per source varies (see Table 7.2), a Kruskal-Wallis H-test reveals these differences are not statistically significant. Prior research indicates that personally *asking* others to contribute would yield the highest accuracy per visit. While visits sourced from asking were the most accurate, they were relatively similar to dynamic, the 2nd highest accuracy sourcing method per visit. Compared to prior work manually asking individuals to contribute [29], the dynamic sourcing methods of CS Open Rankings and Databaits generated highly accurate edits without requiring manual intervention or messaging individuals. These automated dynamic sourcing methods provide evidence for the type of features to enable continuous data maintenance. Answering the call for future research, we posed in Chapter 5.

Visits from the asking and dynamic sources made contributions with at least 80% accuracy. This minimum accuracy is higher than the 75% accuracy reported in the first version of Drafty in Chapter 3. Notably, the static sourcing method generated the lowest accuracy and number of edits among the four sourcing methods. The static sourcing method consists of making forum posts or creating static analyses that do not update over time (i.e., CS Open Data). Overall, organic visitors generated the highest number of edits and did so at an accuracy comparable to Chapter 5 or better than Chapter 3. However, given the ability, creating a dynamic sourcing method is recommended as it will generate more accurate edits compared to the other methods we explored. If edits are required quickly, having one person ask another to contribute provides an excellent method to target specific contributions.

Edits per Visit

This analysis compares the conversion rate and average edits per source (organic, dynamic, static, and asking). Counting the visits from the organic source, 198 made at least one edit, and 3,408 did not, for a conversion rate of 5.8%. Counting the visits from the dynamic source, 72 made at least one edit, and 1,402 did not, for a conversion rate of 5.1%. Counting the visits from the static source, 84 made at least one edit, and 1,785 did not, for a conversion rate of 4.7%. Counting the visits from the asking source, 51 made at least one edit, and 387 did not, for a conversion rate of 13.2%. A chi-squared test reveals these frequencies were significantly different, $X^2(3, N = 6982) = 43.6$, $p < 0.001$. The following post hoc comparisons between sources use the Benjamin-Hochberg method to control for False Discovery Rate [216]. The reported increases in the likelihood of making an edit per visit compare each source's conversion rate. Visits from the asking source are 2.3 times more likely to make an edit than visits from the organic source to make more edits per visit. A post hoc chi-squared test reveals increase is statistically significant, $X^2(1, N = 3795)$ = 29.59, $p < 0.001$. Visits from the asking source are 2.6 times more likely to make an edit than visits from the asking source. A post hoc chi-squared test reveals increase is statistically significant, $X^2(1, N = 1789)$ = 29.40, $p < 0.001$. Visits from the asking source are 2.8 times more likely to make an edit than visits from the static source. A post hoc chi-squared test reveals increase is statistically significant, $X^2(1, N = 2172)$ = 37.73, $p < 0.001$. These results show that asking visitors to contribute data increases conversion rate and mirrors previous research comparing asking people to contribute to altruistic causes versus bots [200]. However, visits from the asking source made less than a third of the edits compared to visits from the organic and dynamic sources (see Table 7.2).

A Kruskal Wallis H-Test reveals the average edits per visit between the sources (organic, dynamic, static, and asking) were significantly different, $X^2(3, N = 6982) = 45.5$, $p < 0.001$. The reported increases in the likelihood of making an edit per visit compare the average edits per visit per source. Visits from the asking source (M = 1.11 SD = 5, N = 387) made 2.5 times more edits per visit than visits from the organic source (M = 0.44 SD = 5, N = 3408). A Mann-Whitney U test reveals this increase is statistically significant, $U = 609245$, $p < 0.001$. Visits from the asking source (M = 1.11 SD = 5, N = 387) made more edits per visit than visits from the organic source (M = 0.87 SD = 8, N = 1402). A Mann-Whitney U test reveals this 21.5% increase in edits per visit is statistically significant, $U = 249798$, $p < 0.001$. Visits from the asking source (M = 1.11 SD = 5, N = 387) made 7.1 times more edits per visit than visits from the static source (M = 0.16 SD = 1, N = 1785). A Mann-Whitney U test reveals this increase is statistically significant, $U = 375733$, $p < 0.001$. While visits from the dynamic source made 2 times more edits per visit compared to organic visits and 5.6 times more edits compared to visits from the static source, Mann-Whitney U tests revealed these increases were not statistically significant. Thus, while asking people might be beneficial for targeting specific data [28, 200], it will likely not generate enough edits to maintain a dataset, especially considering the time it takes to email or tweet individuals asking them to contribute. Thus, building features to attract visits automatically is still vital to maintaining public data [224]. Understanding how to recruit people to maximize contributions can help data curators focus their time on efforts that produce the most contributions.

Engagement per Visit

This section compares the average interactions per visit, session length in seconds per visit, and engagement score per visit between the four sources (i.e., recruitment methods) from 2022: organic, dynamic, static, and asking. All post hoc comparisons between sources use the Benjamin-Hochberg method to control for False Discovery Rate. Table 7.3 shows that while visits from the dynamic source had the lowest average session length, interactions, and engagement score per visit, visits from the dynamic source were engaged when comparing these values per second. Given the high conversion rates and accuracy from the dynamic and asking sources, some of these visits likely arrived at Drafty, edited data quickly, and left rather than lurk and explore. This observation of low session times compared to contribution behaviors is contrary to prior research on Wikipedia [78].

	Visits	**Avg Session Length per Visit**	**Avg Interactions per Visit**	**Avg Engagement Score per Visit**
organic	3408	7 mins 40 secs	0.99	42.1
dynamic	1402	3 mins 14 secs	0.71	31.3
static	1785	4 mins 7 secs	0.38	47.3
asking	387	3 mins 56 secs	0.33	48.8

Table 7.3: This table shows the average interactions, session length, and engagement score per visit. Organic visits had the highest average session length. At the same time, visits from the dynamic source had the lowest average number of interactions per visit despite having 99.1% accuracy per visit. These numbers indicate that visits with the highest average accuracy (dynamic and asking) had the shortest sessions. This indicates that many of these visitors were likely editing and leaving versus lurking and exploring the data.

This result compares the average interactions per visit per source. A Kruskal Wallis H-Test reveals the average interactions per visit between the sources (organic, dynamic, static, and asking) were significantly different, $X^2(3, N = 6982) = 660.4$, $p < 0.001$. However, when splitting visits by their source (organic, dynamic, static, and asking), Pearson's Correlation reveals no effect between accuracy and the average interactions per visit. Therefore, while average interactions were different per source, this did not affect the accuracy of edits. This result mirrors prior sections' results finding no relationship between the number of interactions and accuracy.

Average Interactions per Visit					
	Avg	organic	dynamic	static	asking
organic	35.3	-	48.5% ↑	8.1% ↓	11.4% ↓
dynamic	23.8	32.7% ↓	-	40.3% ↓	38.1% ↓
static	39.8	12.8% ↑	1.7 times ↑	-	3.7% ↑
asking	38.4	8.8% ↑	1.6 times ↑	3.5% ↓	-

Table 7.4: This table shows the effect sizes for the average interactions per visit between different sources (i.e., recruitment methods). The reported effect sizes compare the left-most column with the source from the top row. Mann-Whitney U tests reveal all reported differences are statistically significant, $p < 0.001$ in all comparisons. Post hoc comparisons use the Benjamin-Hochberg method to control for False Discovery Rate.

The session length per visit per source can help indicate how long different visitors spent on Drafty. A Kruskal

Wallis H-Test reveals the average session length per visit between the sources (organic, dynamic, static, and asking) were significantly different, $X^2(3, N = 6982) = 671.9$, $p < 0.001$. Table 7.5 shows these differences were significant across different sources. However, when splitting visits by their source (organic, dynamic, static, and asking), Pearson's Correlation reveals no effect between accuracy and the average session length per visit. Thus, while session lengths were different per source, this did not affect the accuracy of edits. This result indicates visitors to Drafty can make highly accurate edits without needing an extended time to acclimate to the system.

Average Session Length per Visit					
	Avg	organic	dynamic	static	asking
organic	460 seconds	-	2.4 times ↑	1.9 times ↑	2.0 times ↑
dynamic	194 seconds	57.8% ↓	-	21.4% ↓	17.8% ↓
static	247 seconds	46.4% ↓	27.2% ↑	-	*4.4% ↓
asking	236 seconds	48.7% ↓	21.6% ↑	*4.6% ↑	-

Table 7.5: This table shows the effect sizes for the average session length in seconds per visit between different sources (i.e., recruitment methods). The reported effect sizes compare the left-most column with the source from the top row. Mann-Whitney U tests reveal all reported differences (*except asking versus static) are statistically significant, $p < 0.001$ in all comparisons. Post hoc comparisons use the Benjamin-Hochberg method to control for False Discovery Rate.

The engagement score per visit per source can help indicate how visitors were engaged with Drafty. A Kruskal Wallis H-Test reveals the average engagement score per visit between the sources (organic, dynamic, static, and asking) were significantly different, $X^2(3, N = 6982) = 372.8$, $p < 0.001$. When splitting visits by their source (organic, dynamic, static, and asking), Pearson's Correlation reveals an effect per source between accuracy and the average engagement score per visit. There is a small effect within the organic source (r = 0.14, p ¡ 0.001). There is a medium effect within the dynamic source (r = 0.36, p ¡ 0.001). There is a small effect within the static source (r = 0.10, p ¡ 0.001). There is a small effect within the asking source (r = 0.15, p ¡ 0.001). While Table 7.6 shows that visits from the dynamic source had the lowest engagement score, a subset of these visits were highly engaged and accurate. The dynamic sources (with over 1400 visits) were able to attract a subset of highly engaged visitors. In contrast, the organic and static sources attracted visitors who lurked for extended periods without contributing to Drafty's data.

Average Engagement Score per Visit					
	Avg	organic	dynamic	static	asking
organic	42.1	-	34.5% ↑	11.0% ↓	13.8% ↓
dynamic	31.3	25.7% ↓	-	33.9% ↓	35.9% ↓
static	47.3	12.4% ↑	1.5 times ↑	-	3.2% ↑
asking	48.8	16.0% ↑	1.6 times ↑	3.1% ↓	-

Table 7.6: This table shows the effect sizes for the average engagement score per visit between different sources (i.e., recruitment methods). The reported effect sizes compare the left-most column with the source from the top row. Mann-Whitney U tests reveal all reported differences are statistically significant, $p < 0.001$ in all comparisons. Post hoc comparisons use the Benjamin-Hochberg method to control for False Discovery Rate.

7.5.7 Visitors who Created Insights using Did You Know were More Accurate and Engaged

This section compares the dependent variables per visit for those who freely created an insight within Drafty versus those who did not. Insights from Databaits can encourage engagement and contributions by providing visitors autonomy to interact with Drafty's tabular data. The Databaits API's automatically generated insights empower the "Did You Know" feature to encourage reciprocity between Drafty's data and its contributors.

Accuracy per Visit

Among visits in 2022 with at least one edit checked for correctness, the average accuracy for visits that created an insight (M = 99% SD = 0.12, N = 73) versus visits that did not (M = 94% SD = 0.23, N = 154) was 5 points higher. A Mann-Whitney U test reveals this 5% increase in average accuracy per visit is statistically significant, $U = 6091$, $p = 0.03$. The Databaits API and Did You Know were deployed in Drafty in the Spring of 2022. This is in response to years of people creating static insights from Drafty's data. The idea was inspired by developing the public data system Sketchy in Chapter 6. Chapter 6 showed initial evidence of providing real-time features within public systems that derive insights from its data to motivate engagement and contributions. One in three visitors who edited data on Drafty also freely chose to create insights in the Did You Know Modal. At a minimum, visitors that create insights in Drafty are also highly accurate. This shows that dynamic features that update in real-time based on evolving data can increase user agency and help engage and elicit highly accurate contributions. This is an essential outcome for a public data system like Drafty to help maintain an accurate dataset over time. This relationship between accuracy and visits creating insights could help predict accurate visits in the future.

Edits per Visit

This analysis compares the number of visits in 2022 with at least one interaction that freely created insights in the Did You Know Modal versus those that did not. Among visits that created an insight, 105 visitors made at least one edit, and 275 did not, for a conversion rate of 27.6%. Among visits that did not create an insight, 301 visitors made at least one edit, and 6,302 did not, for a conversion rate of 4.6%. A chi-squared test reveals these frequencies were significantly different, $X^2(1, N = 6983) = 345.1$, $p < 0.001$. Notably, the conversion rate of visits that created an insight was 6 times higher than visits that did not create an insight. Among all our results, visits that created insights had the highest conversion rate. On average, visits that created an insight (M = 2.81 SD = 11.66, N = 380) made 7.8 more edits per visit than visits that did not (M = 0.36 SD = 4.17, N = 6603). A Mann-Whitney U test reveals this increase in edits per visit is statistically significant, $U = 1546049$, $p < 0.001$. These results indicate that visits creating insights contribute data without the system nudging them. This result contradicts prior work showing how nudging people can elicit contributions [200, 225]. When visits create an insight, this serves as a strong indicator they will edit and contribute data to Drafty. Thus, providing a feature that can be used to identify visits that will contribute to maintaining Drafty's data.

Engagement per Visit

While spreadsheets are a ubiquitous interface for viewing and editing tabular data for many professions, they can lack features that drive user engagement. Insights generated by the Databaits API, presented to the user as a "Did You Know", are designed to give Drafty's visitors real-time insights about its data. This analysis compares the average interactions, visit length in seconds, and user engagement profile per visit from visitors who created an insight using the Did You Know Modal versus those who did not create an insight in 2022.

Visits that created an insight (M = 58 SD = 100, N = 380) made around 25 more interactions per visit compared to visits that did not create an insight (M = 33 SD = 65, N = 6603). A Mann-Whitney U test reveals this increase of visits created an insight making 1.8 times more interactions per visit is statistically significant, $U = 1619791$, $p < 0.001$. The average session length per visit for visits that created an insight (M = 545 SD = 852, N = 380) compared to visits that did not (M = 328 SD = 838, N = 6603) was around 3 minutes and 38 seconds longer per visit. A Mann-Whitney U test reveals that the session length of visits being 2.2 times longer among visitors who created an insight is statistically significant, $U = 1750462$, $p < 0.001$. The average engagement score for visits that created an insight (M = 78 SD = 117, N = 380) compared to visits that did not (M = 40 SD = 69, N = 6603) was around 38 points higher. A Mann-Whitney U test reveals that the engagement score being 2 times larger for visits that created an insight is statistically significant, $U = 1744716$, $p =< 0.001$.

These results indicate that highly engaged users also used the Did You Know feature to create new insights within Drafty. Providing Did You Know as a feature provided another method to help engage and potentially keep visitors using Drafty, replicating results from Skethcy in Chapter 6. Even if it only appeals to the most engaged users, it provides a valuable tool to keep these visitors on Drafty. The following section analyzes visitors' next actions within Drafty after creating or seeing an insight in the Did You Know Modal to understand better if their behaviors benefit Drafty's goal of maintaining evolving data.

7.5.8 After Creating or Seeing an Insight in the Did You Know Modal: What were Visitors' Next Actions?

Drafty presents visitors with an insight from the Databaits API in a modal window (see Figure 7.3 for an example of the Did You Know Modal) as part of a confirmation and thank you message after someone edits a cell, adds a new row, or deletes an existing row of data. Visitors can also freely choose to create an insight within Drafty's Did You Know Modal. During the designated study period in 2022, visitors saw an insight 902 times while they freely chose to create an insight 1,542 times.

Overall, after the Did You Know Modal is presented to a visitor, there is a 46% chance they choose to create another insight within the modal. Given a visitor will eventually have to either leave or close the modal, this result shows how highly engaged visitors are with insights inside the modal. In the Did You Know Modal, a visitor can choose to create an insight that is "similar" or "random" compared to the current insight. Prior research on recommender systems shows users prefer to see similar recommendations to examples they have seen before [100, 245]. Overall in Drafty, visitors were 6 times more likely to create a "random" Databait than a "similar" Databait. A one proportion z-test reveals that 86% of the time visitors choose to create a "random" Databait compared to a "similar" Databait within the modal is statistically significant, $z = 23.86$ $p < 0.001$. Not only does this result run contrary to prior research in recommendation systems [77], the interface design of the Did You Know Modal always presents the "similar" option first, thus influencing the visitor to select the "similar" option [36]. Given these design choices and results, it provides strong evidence Drafty's visitors preferred to create randomized examples from the source dataset in the form of insights created by the Databait API.

To examine these results further, this section features two sets of analysis using two-tailed McNemar's tests, with the continuity correction for all proportions, to study visitors' next actions taken within the Did You Know Modal and after the Did You Know Modal is closed. These analyses compare visitors' next action within Drafty after they freely choose to "create" or "see" (i.e., post edit) an insight within Drafty.

Visitors who freely create an insight are more likely to continuously engage with insights within the modal window compared to visitors who see them post edit. For example, within the Did You Know Modal, visitors could perform one of seven actions listed in Table Table 7.7. Notably, visitors who saw an insight post edit were 6.6 times more likely to create a "random" insight than a "similar" insight. A one proportion z-test reveals this increase is statistically significant, $z = 7.56$ $p < 0.001$. Also, visitors who create an insight were 5.8 times more likely to create a "random" Databait than a "similar" Databait. This mirrors the prior result even when splitting visitors based on why they saw the Did You Know Modal. This result shows visitors preferred to see "random" insights. A one proportion z-test reveals this increase is statistically significant, $z = 23.13$ $p < 0.001$.

When visitors created an insight in the Did You Know Modal, there was a 66.2% they would create another insight as their next action within the modal. This probability is 5.6 times compared to when visitors see an insight post edit. A McNemar test shows this increase was statistically significant $\chi^2(1) = 106$, $p < 0.001$. There is a trend of visitors creating insights in the Did You Know Modal, compared to those who see an insight post edit. When visitors saw an insight post edit, there was a 78.8% chance they would close the Did You Know Modal as their next action. This probability is 3.8 times greater than visitors who created an insight. A McNemar test shows this increase was statistically significant $\chi^2(1) = 711$, $p < 0.001$. While visitors who saw an insight post edit were more likely to close the modal, 68.7% of them contributed Drafty after closing it. After the Did You Know Modal is closed, a visitor's next actions align with one of three outcomes listed in Table Table 7.8. Overall, a visitor is just as likely to leave Drafty after

		Create a Databait					
Freely chose to create the Databait	**Tweet Databait**	**Total***	*Create Similar*	*Create Random*	**Leave Drafty**	**Close Modal**	**Search Drafty**
Yes	1.0%	66.2%	9.7%	56.5%	11.3%	20.9%	-
No	1.7%	11.8%	1.6%	10.5%	7.8%	78.8%	-
Likelihood	0.6	5.6	6.2	5.5	1.5	0.3	-

Table 7.7: Likelihood compares the next actions of visitors who freely chose to create a Databait with those who did not (i.e., they saw the Databait Modal after a contribution) within the Databait Modal. This data is from the 2022 Study period. The column "Create* Databait" combines the data from the "Create Similar" and "Create Random" columns. Visitors who freely chose to create a Databait were most likely to create another Databait as their next action. The most common action for visitors who saw a Databait after making an edit was to close the Databait Modal. McNemar tests show all relationships were statistically significant $p < 0.001$. There are no instances of these visitors selecting a value from the Databait to "search" Drafty's dataset. This feature might have been difficult to discover.

creating an insight versus seeing one. However, visitors who did not freely choose to create an insight were twice as likely not to contribute to Drafty within that same visit. A McNemar test shows this decrease was statistically significant $\chi^2(1) = 182$, $p < 0.001$.

		Interacted after Databait Modal	
Freely chose to create the Databait	**Leave**	**No Contribution**	**Contributed**
Yes	11.4%	10.2%	78.3%
No	11.1%	20.2%	68.7%
Likelihood	1.0	0.5	1.1

Table 7.8: Likelihood compares the next actions of visitors who freely chose to create a Databait with those who did not (i.e., they saw the Databait Modal after a contribution) within the Databait Modal. This data is from the 2022 Study period. Visitors who did not freely chose to create a Databait were twice as likely to not make a contribution to Drafty after the Databait Modal. McNemar tests show all relationships were statistically significant $p < 0.001$.

Drafty's Visitors Prefer Randomized Examples

In a public data system like Drafty, where visitors share common interests, they prefer to view randomized examples from the data compared to similar examples. This finding simplifies how to recommend examples when a system's users share interests. Recommendation or example generation features should focus on generating many possible examples from the data. This finding mirrors prior research in Chapter 6 that indicates users might prefer randomized examples from a dataset of evolving sketches [223].

7.6 Discussion

The discussion focuses on the importance of using low-effort dynamically regenerated sources to attract everyday people to visit and contribute to public data systems like Drafty. It also discusses how these contributions help maintain evolving data and how accurate evolving data will enable AI over time to respond to these changes in evolving information.

7.6.1 Dynamic Sources and Automatic Insights to Attract and Engage Users

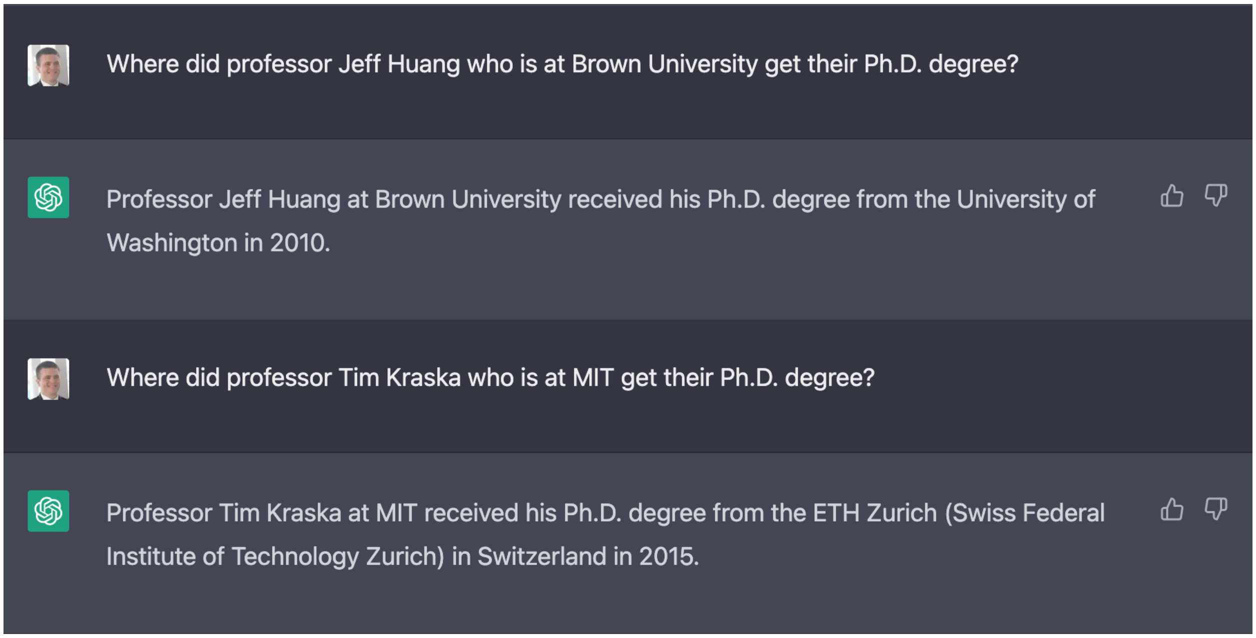
Our log-based and descriptive analysis reveals how the new features developed for 2022 increased accuracy, edits, and user engagement per visit within a real-world scenario. This descriptive approach follows the recommendation to study natural behavior in social computing systems [19]. Understanding who will contribute is a long-standing research question [242]. Simultaneously, evaluating multiple methods to attract visitors delivers insights into their motivations in the wild. Attracting users using dynamic sources that automatically update, such as CS Open rankings and Databaits, can produce relatively equal numbers of accurate contributions as manually asking people to contribute data. The long-term benefit of these dynamic sources is that they require little maintenance and can continuously attract and motivate contributions from everyday visitors to a public data system. This allows the curators and system builders to spend their time and energy elsewhere rather than manually engaging the community of users.

One area of future research is how these small niche communities within a system like Drafty compare to large communities on popular platforms such as Wikipedia or Wikidata. While our results show how Drafty's small community of like-minded users possesses a mutual interest and makes highly accurate contributions, Drafty has yet to become popular to the point where it attracts many contributors external to the existing community. Unlike Wikipedia and Wikidata, Drafty provides a quick, low-effort interface to enable new and returning visitors to make accurate contributions [131]. Our results show that building small research systems is a valuable method to help people while delivering broader impacts compared to pushing all the data to large platforms such as Wikipedia or Wikidata. Advertising a public data system as a product is simpler than a single Wikipedia page. Viewing a public data system, such as Drafty, as a product with anonymous customers clarifies how these findings translate beyond sociotechnical systems and veer towards marketing, economics, and computational social theories.

7.6.2 Maintaining Data Quality to Enable AI

This longitudinal research effort presents a solution to maintain evolving data over time. Providing automatic insights that are dynamically regenerated can motivate and engage a community of people to make accurate contributions. The benefit of employing such methods enables the long-term maintenance of evolving information that can power many solutions and models to help close the loop on developing and deploying naturalistic machine learning methods.

Many large language models are trained using the history of the World Wide Web. Present day, the answers to their prompts appear relevant. However, much of the knowledge and subjective opinions curated by people evolve. For example, Figures 7.6 and 7.7 show how ChatGPT version 3 can validate some of Drafty's older Computer Science professor data, but it lacks more recent data from the past year. Drafty and its anonymous community still play a vital



Where did professor Jeff Huang who is at Brown University get their Ph.D. degree?

Professor Jeff Huang at Brown University received his Ph.D. degree from the University of Washington in 2010.

Where did professor Tim Kraska who is at MIT get their Ph.D. degree?

Professor Tim Kraska at MIT received his Ph.D. degree from the ETH Zurich (Swiss Federal Institute of Technology Zurich) in Switzerland in 2015.

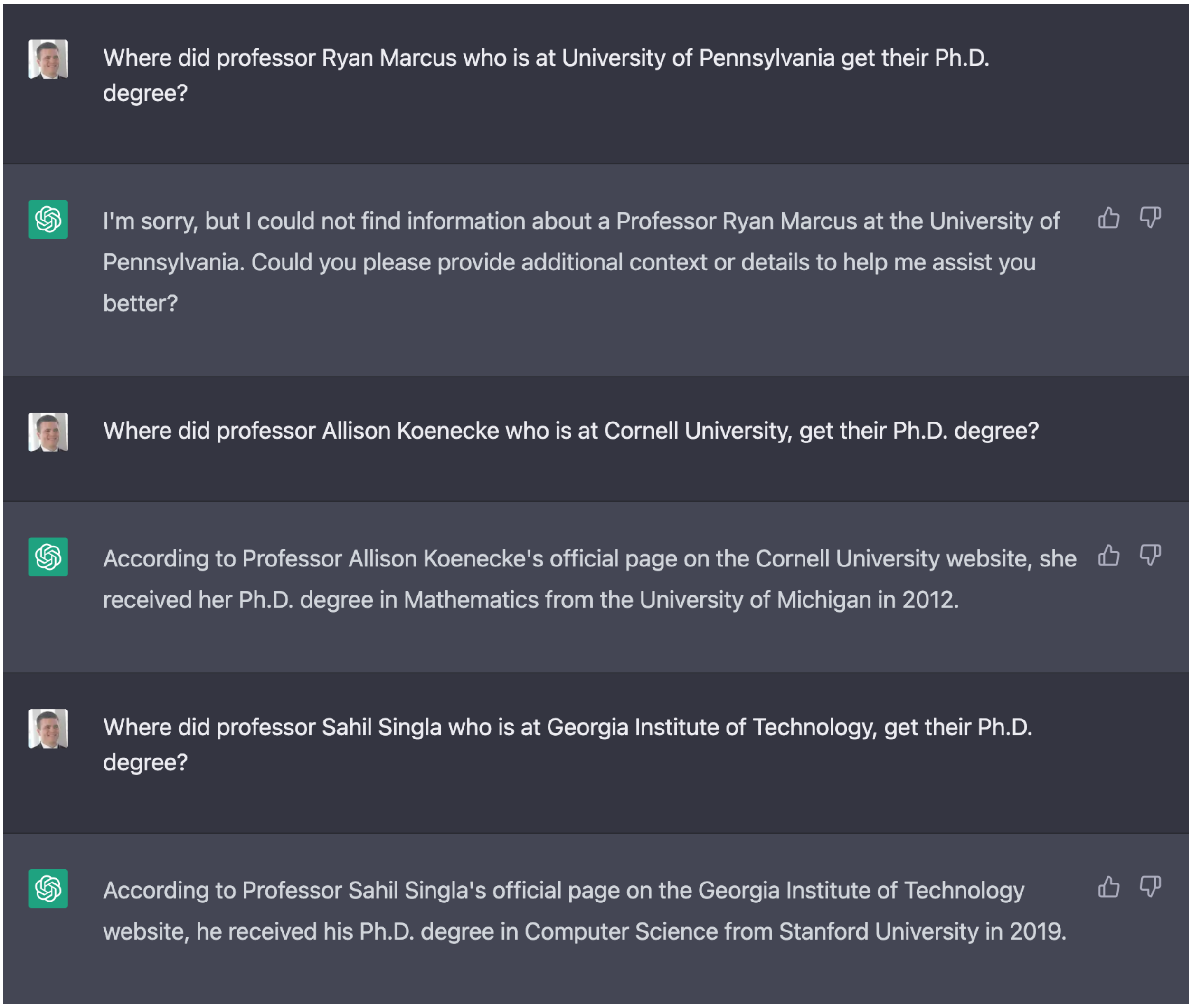
Figure 7.6: The university granting each Ph.D. is correct, while the year their degrees were granted is incorrect. However, for professor Jeff Huang, ChaptGPT did provide 100% correction information prior to March 14th.

role in maintaining evolving data. Future research should compare large language models, anonymous everyday visitors as unpaid contributors, and paid crowdworkers on their ability to maintain evolving data in a system such as Drafty.

To develop these models, autonomous systems and methods are trained on large data sets and can produce intelligent and engaging responses [210]. There is a role for human-centric systems like Drafty to fulfill in the future. Public data systems, like Drafty, decompose complex tasks into simple contributions. This task decomposition is similar to piecework and crowdsourcing [5]. Building on these ideas, Drafty's users are the stewards of their community's information and interests. By centralizing data under a common theme and providing it publicly to a community of like-minded users, they will maintain its integrity over time by structuring information, tables, and other easily accessible interfaces. It will make extracting insights to create training data for large models easier. If we want to continue integrating AI into our society, we must also understand how to maintain the quality of its training data over time. While Drafty plays a small role in this burgeoning need, future research can help answer this call by developing small bespoke public data systems to engage communities of users. This longitudinal research combines social theories, human-computer interaction, and data science to hopefully deliver beneficial real-world outcomes.

7.7 Limitations

This initial set of Databaits sentence templates is only in English. It is essential future researchers consider translating sentence templates into other languages. We conducted several pilot studies developing the sentence templates to



Where did professor Ryan Marcus who is at University of Pennsylvania get their Ph.D. degree?

I'm sorry, but I could not find information about a Professor Ryan Marcus at the University of Pennsylvania. Could you please provide additional context or details to help me assist you better?

Where did professor Allison Koenecke who is at Cornell University, get their Ph.D. degree?

According to Professor Allison Koenecke's official page on the Cornell University website, she received her Ph.D. degree in Mathematics from the University of Michigan in 2012.

Where did professor Sahil Singla who is at Georgia Institute of Technology, get their Ph.D. degree?

According to Professor Sahil Singla's official page on the Georgia Institute of Technology website, he received his Ph.D. degree in Computer Science from Stanford University in 2019.

Figure 7.7: Professor Ryan Marcus announced he was joining Penn in the spring of 2022 on Twitter. The university granting the Ph.D.'s for Allison and Sahil are correct, while the year their degrees were granted is incorrect.

find the exact wording that people anecdotally found the easiest to understand. We believe translating these sentence templates will require many careful considerations and possibly entirely new sentence templates to account for differing grammar and connotations within other languages.

Another limitation of the current work is the tabular dataset used for the main study focuses only on Computer Science professors. Our development of the Databaits API was motivated by this particular dataset containing evolving information that requires people to make edits over time. Hence, our initial motivation was to 1) develop a reward mechanism to motivate users to make additional unpaid contributions; and 2) create something interesting to attract new and existing users to visit the tabular dataset. In the future, we can develop additional tabular datasets with evolving information to understand how Databaits and its automatic insights scale across different data types. Our initial Databaits Validation Study shows initial evidence this should be true.

7.8 Conclusion

This chapter shows how developing features within the system to reduce the estimated effort to contribute can help increase the number and accuracy of contributions to Drafty compared to older versions of the system from Chapters 3 and 5. CS Open Rankings also serves as an extension to Drafty that provides immediate utility to Drafty's community. Thus, every time Drafty's visitors contribute accurate data, they are also indirectly helping their peers and others they might not know yet.

This chapter develops a new version of the public data system Drafty to engage and attract organic visitors to become contributors. During the first year of deployment, CS Open Rankings and the Databaits API were developed using Drafty's data of computer science profiles as a source to generate automatic insights dynamically. The goal of these additional systems is to automatically and continuously attract and engage visitors who make accurate contributions with little to no long term effort from the data curators. Results show that visitors from these organic sources were as accurate as those who were personally asked, using friendsourcing [28], without engaging in long term negative persuasive behaviors like nudging. Insights from the Databaits API, presented to users as a "Did You Know", elicited more contributions and higher engagement over time. Notably, visitors preferred to see a random insight, or "Did You Know", compared to a similar one. Thus, demonstrating how future recommendation systems can employ randomization to appeal to the preferences and behaviors of anonymous communities of like-minded users.

This chapter demonstrates how engaging users with automatic insights incentivizes participation and contributions to a public data system. Maintaining a public data system's evolving data enables reciprocity between the data and insights a system presents and a visitor seeks. This automated loop integrates real-world observations, social theories, and human-data interactions to deliver this solution. Deploying public data systems, like Drafty, will enable future data curators to maintain evolving data to meet the information and data demands of users and AI alike.

Chapter 8

Towards Fair & Equitable Incentives to Motivate Crowd-Contributions

This chapter presents a survey study to understand the crowd contribution preferences of paid crowdworkers and unpaid contributors to Drafty. This chapter features a discrete choice experiment, where Drafty's visitors choose their preference between two hypothetical crowd contribution tasks. The answers to these pairwise comparisons produce utility scores across various attributes and their levels that describe what motivates someone to prefer a crowd contribution task. For example, two attributes are pay level per hour and estimated time to complete. The analysis uncovers the relationships to describe the trade-offs between pay level, estimated time to complete, task difficulty, various task requirements, and user perceptions. I am solely responsible for all of this chapter's writing and analysis. My co-authors from the submission have provided guidance and suggestions to the initial writing.

8.1 Introduction

As part of the crowd, people contribute their time and knowledge to research efforts and datasets for various reasons. These crowd contributions can take various forms, from completing surveys or editing data on Wikipedia, Google Spreadsheets, or even public data systems such as Drafty. The choice someone makes to visit a public data system can differ from their choice to contribute. People contribute to research efforts and datasets for various extrinsic and intrinsic reasons [159, 160]. Turning visitors who do not contribute (i.e., lurkers) into contributors is a complex, multi-faceted scenario [18].

Researchers and practitioners constantly balance task design, motivational factors, and incentive mechanisms to yield accurate contributions from the crowd [123]. We need to balance task design, motivational factors, and incentive mechanisms to elicit contributions. When building systems to run crowdsourcing or peer production tasks to elicit crowd contributions, assessing how these systems provide fair and equitable incentives to motivate crowd contributions can be challenging.

Someone who makes crowd contributions, for example, a paid crowdworker on Prolific or a Wikipedia editor, chooses different tasks based on different attributes: payment (or lack thereof) [31, 73], time to complete [46], task difficulty [98, 122], and task perception (i.e., their level of interest, does the task help others) [159, 160]. They make trade-offs between these attributes when choosing how to contribute their time. Understanding these trade-offs is essential to understanding crowd contributors' motivations and developing fair incentive mechanisms to reward their contributions. The crowd contributors can be unpaid contributors or paid crowdworkers completing tasks and contributing data.

One approach to studying these trade-offs is a discrete choice experiment. Discrete choice experiments are a quantitative method to analyze user preferences [128]. In a discrete choice experiment, users choose between two hypothetical alternative scenarios. Each realistic scenario contains a set of attributes, and each attribute has a set of labels. For example, in Table 8.1, which of these two tasks with attributes [and levels]: payment per hour [$0.00 vs. $12.00], time to complete [5 vs. 15 minutes], task difficulty [hard vs. easy], and level of interest [high vs. low] would someone choose to complete?

	Task A	**Task B**
Payment per Hour	$0.00	$12.00
Time to Complete	5 minutes	15 minutes
Task Difficulty	hard	easy

Table 8.1: A simple example of a choice set featuring two hypothetical tasks a crowd contributor could select. Choice sets are used in a discrete choice experiment to elicit user preference. Users would select Task A or Task B to indicate their stated preference. See Figure 8.1 for an actual choice set used in the survey study.

These answers can create utility functions to evaluate the trade-offs between and within the attributes and levels. Researchers, practitioners, and system builders can use these utility functions to develop fair and equitable rewards and incentive mechanisms in real-world crowd-powered and peer production systems.

This chapter is a survey-based study that develops a discrete choice experiment to analyze the trade-offs Drafty's everyday visitors make when choosing to make crowd contributions. Drafty is a real-world public data system of computer science faculty profiles where anyone can freely visit and edit its data. Drafty's visitors can be unpaid contributors, similar to Wikipedia editors, or paid crowdworkers from Prolific editing tabular data using the Drafty spreadsheet interface. This chapter seeks to provide evidence of what motivates crowd contributors to contribute.

The research asks the following questions:

1. What attributes and levels to design fair and equitable incentive mechanisms for crowd contribution tasks apply

to a public data system's paid crowdworkers and unpaid contributors?

2. What attributes and levels should be individuated per group of users (paid crowdworkers vs. unpaid crowdworkers)?

The results from this chapter show that among paid crowdworkers and unpaid contributors, the pay level, estimated time to complete, and user's perception of a crowd contribution task affect their preferences to complete a task. However, while pay level is most important to paid crowdworkers, task perception is most important to unpaid contributors. Results show if you are designing a fair and equitable system to collect crowd contributions, it is essential to develop tasks and features where users interact with others and complete tasks that are interesting, ethical, quick, and easy to complete. These contributions will help others and are owned by a public community or the user themselves.

8.2 Method

The methods section covers the study design considerations, terminology, and process for developing and running a survey-based study featuring a discrete choice experiment with Drafty's users in the wild.

8.2.1 Study Design Considerations

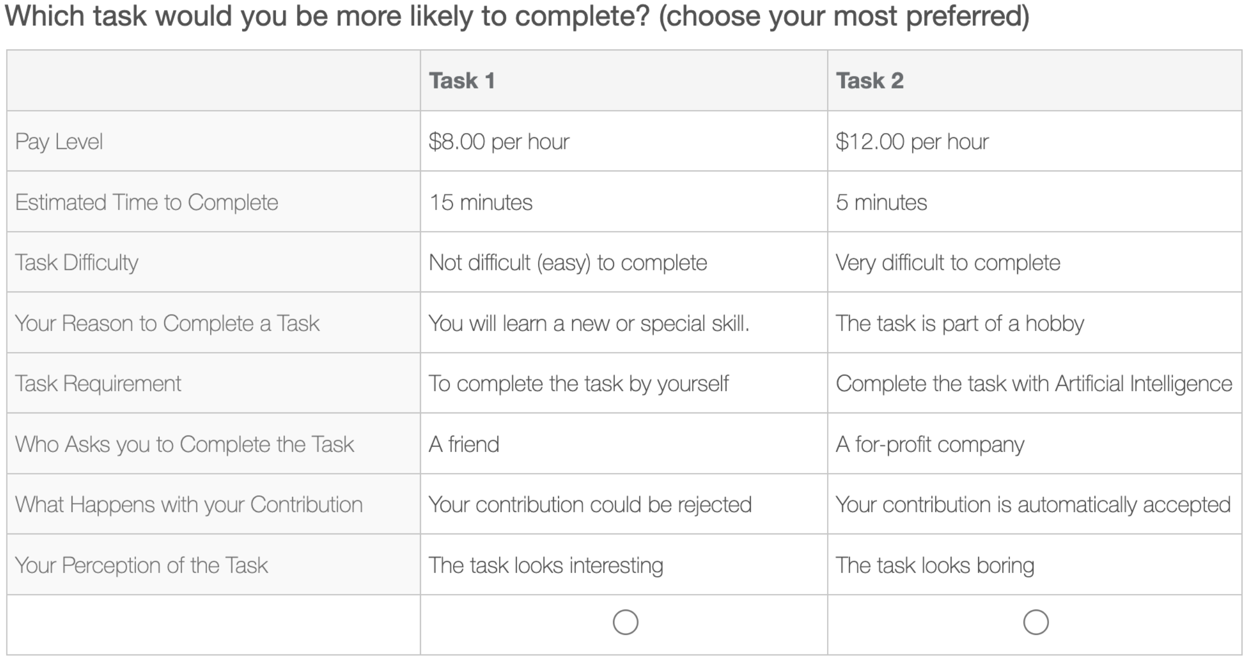
Designing a discrete choice experiment is a sequential procedure with many well-defined steps, from selecting attributes and associated levels and their subsequent choice sets to pilot studies [149]. Reviewing related crowdsourcing and peer production literature combined with previous chapters' findings will yield the set of attributes and levels to create a discrete choice experiment. We will use these initial attributes and levels to conduct a pilot study by recruiting paid crowdworkers and unpaid contributors within Drafty. The findings from a preliminary study will inform the design of a final discrete choice experiment to help understand the trade-offs users make across various attributes and associated levels that comprise crowd contribution tasks.

Study Design Terminology

Discrete choice experiments are a research method commonly used in Healthcare, Health Informatics, Business, and Economics research. This section covers the basic terminology and methods used in a discrete choice experiment.

Crowd contribution task is a task where a person interacts with a computer or device to contribute information or data. This person can be paid or unpaid for the completion of this task. For example, people contributing to Wikipedia would be a crowd contribution task. A crowd contribution task could also be a paid crowdworker from Prolific completing a survey about technology use. Whether paid or unpaid, all visitors to Drafty are completing crowd contribution tasks when editing Drafty's data.

Discrete choice experiments are a preference elicitation technique where participants make a choice from two or more hypothetical alternatives, see Figure 8.1. They were originally developed by Louviere and Woodworth for use in health economics and are commonly used in economics, health, and market research [142]. A discrete choice experiment better resembles people's real-world decisions compared to other stated preference methods such as ranking or Likert-scale [149].



Which task would you be more likely to complete? (choose your most preferred)

	Task 1	Task 2
Pay Level	$8.00 per hour	$12.00 per hour
Estimated Time to Complete	15 minutes	5 minutes
Task Difficulty	Not difficult (easy) to complete	Very difficult to complete
Your Reason to Complete a Task	You will learn a new or special skill.	The task is part of a hobby
Task Requirement	To complete the task by yourself	Complete the task with Artificial Intelligence
Who Asks you to Complete the Task	A friend	A for-profit company
What Happens with your Contribution	Your contribution could be rejected	Your contribution is automatically accepted
Your Perception of the Task	The task looks interesting	The task looks boring
	○	○

Figure 8.1: An example of two hypothetical alternative crowd contribution tasks presented to survey participants in Qualtrics. Sometimes this is referred to as a choice set. The attributes are on the leftmost column, and their associated levels are under the columns labeled "Task A" and "Task B."

Choice Set is at least two hypothetical alternatives where each alternative has the same attributes but different associated levels per attribute. Each alternative, or choice set, consists of at least two attributes, and each attribute has at least two levels. By presenting participants with a series of hypothetical scenarios (i.e., choice sets), where they are asked to select their preferred option from two or more alternatives that differ in specific attributes or characteristics. By varying each attribute's levels, estimating the relative importance and trade-offs people make during their decision-making process is possible. The choice someone makes when picking between two hypothetical alternatives are the dependent variables in a discrete choice experiment.

Attributes are the independent variables that are being tested [149]. For example, a crowd contribution task's attribute could be pay level, estimated time to complete, and perceived difficulty. Another example is where a box of cereal's attributes could be price, amount of sugar, and brand name. Attributes are often identified by reviewing related research and real-world observations [52].

Levels are an attribute's options, increments, or possible values. They can be continuous, ordinal, or binary.

Prior research often selects levels that are considered realistic because they reflect values people encounter in the real world [52]. For example, Prolific enforces a minimum pay level per hour of $8, while their recommended pay level is $12 per hour. Building on this idea of pay level, normal everyday visitors to Drafty (i.e., unpaid contributors) make contributions for a pay level of $0 per hour. The analysis of a discrete choice experiment provides utility scores to understand what levels negatively and positively affect someone's preference for a given scenario.

Designing the Discrete Choice Experiment

In a discrete choice experiment, people select between two hypothetical alternatives. The attributes and their levels used in the main study are described below. These final sets of attributes and their levels were selected after an initial set was selected after reviewing related research and conducting a pilot study in Fall 2022. In this pilot study, paid crowdworkers and unpaid contributors (i.e., unpaid contributors) within Drafty take an initial survey. This survey features MaxDiff questions where participants selected what motivated them most to contribute and what motivated them the least among the levels for a particular attribute. These participants voluntarily chose to take the pilot survey by selecting a blue button in a banner within Drafty; see Figure 8.2. Participants who completed the survey could submit their email for a one in four chance to receive a $25 Amazon gift card as compensation.

After the pilot study, seven attributes were selected to describe crowd contribution task from a user's perspective: pay level per hour, estimated time to complete, task difficulty, their reason to complete a task, what the task requires them to do, who asked them to complete the task, and their perception of the task. The associated levels per attribute are described below. It is important to note that these attributes and levels will shift as people's perspectives change. One goal in selecting the attributes is understanding the trade-offs people make when completing a crowd-construction task.

Pay Level:

1. $0.00 per hour
2. $4.00 per hour
3. $8.00 per hour
4. $12.00 per hour
5. $16.00 per hour

The pay level per task is among the most frequently researched topics in crowdsourcing [122, 152]. Specifically, requesters on paid crowdsourcing platforms find it challenging to assess what is a fair payment [196]. Paid crowdworkers often use resources like TurkerView to view the hourly pay rate a requester offers, and if that rate is fair [199]. While research has suggested alternative payment schemes such as payments in bulk, payment per task is still the most common because it is easily understood [108]. There is a diminishing return in increasing payment and eliciting higher quality contributions [98]. These linearly increasing pay levels can help study what other task attributes and levels need to be present to fairly motivate people when the payment is not high. Most importantly, regarding Drafty unpaid contributors, what motivates them when monetary compensation is absent (i.e., $0.00 per hour)?

Prior research focused on making unpaid contributions for social good [18] or paying intrinsically motivated users [116, 228]. These parameters affect human choice, and while some have been studied, there is a lack of understanding about the trade-offs people make when choosing tasks in crowdsourcing and peer production scenarios.

Pay level is presented per hour instead of a single dollar amount for many reasons. First, the paid crowdsourcing platform Prolific displays a pay level between $8.00 and $16.00 per hour. While a requester may pay over $16.00 per hour, Prolific enforces fair compensation practices by requiring the requester to pay crowdworkers at least $8.00 per hour if the estimated task time is incorrect. Second, most crowdsourcing research papers communicate the payment per hour crowdworkers receive. Lastly, we recruit paid crowdworkers from Prolific, paying them $8.00,$12.00, or $16.00 per hour. This enables us to study how the stated preferences shift among crowdworkers who accept tasks at different payment levels. $4.00 per hour was chosen to specifically represent a value that should be perceived as underpaid by a crowdworker.

Estimated Time to Complete:

1. 1 minute
2. 5 minutes
3. 15 minutes
4. 30 minutes
5. 60 minutes

Predicting the exact time a task takes to complete is difficult for both paid crowdworkers and requesters alike [218]. However, the time to complete a task is often cited as an essential motivator in choosing to complete paid [69] and unpaid [181] tasks alike. Prior research also shows unpaid users are likelier to spend more time on a task than paid crowdworkers [112].

The exponentially increasing intervals between each “estimated time to complete” was chosen to balance the number of levels and increase the difference between the min and max time. The extremes can represent two different possibilities. In 1 minute, it is reasonable to assume someone can visit a public Google Spreadsheet and add one piece of data they already know in a cell. In contrast, in 60 minutes, someone could visit Wikipedia, read one article, search the various facts and sources to verify each is correct, and correct any inaccuracies within the article. The intention is for the "estimated time to complete" attribute to cover this range of times to help future requesters and system designers to understand the impact task completion times can have on someone's willingness to contribute.

Task Difficulty:

1. Not difficult (easy) to complete
2. Moderately difficult to complete
3. Very difficult to complete

How difficult a task appears to complete is one of the most common factors requesters and systems designers try to optimize [101]. Prior research shows that task difficulty relates to the effort required to complete a task [46, 163]. For example, a paid crowdworker could be required to label ambiguous images or interpret subjective data, such as a Professor's research area. When deciding if they should complete a crowd contribution task, they might favor more straightforward tasks. Liu et al. showed that increased task difficulty could adversely increase the time to complete a task [140]. This is notable because many paid crowdworkers are attempting to maximize the amount they are paid over a specific timeframe. By completing easy tasks, they can better optimize their time and compensation. The results for this attribute could help requesters and systems builders spend more time improving the usability and simplicity of their crowd contribution methods and features.

Your Reason to Complete a Task:

1. You might be paid for doing exceptional work
2. Your contribution benefits you personally
3. You will learn a new or special skill
4. You will get a personal recommendation or learn something new about yourself
5. You get reputation points in a system (special badge, points, credit, etc.)
6. The task is part of your job
7. The task is part of a hobby

Providing the option "Your contribution benefits you personally" builds on the findings from Chapters 4 and 5 and prior research [64] showing that people can be intrinsically motivated to contribute because it benefits themselves. Chapter 4 shows evidence that bonuses or other extra payments for exceptional work can motivate accurate contributions. A common practice is to develop gamified systems to elicit crowd contributions, where the system often awards reputation points or badges for participation [153, 157]. Results showing the effectiveness of gamification from prior research are mixed [147]. While Chapters 4 and 5 offer the promise of higher-paying tasks if paid crowdworkers complete an initial set of tasks. The idea a task might mirror a hobby aligns with some of the motivation from learnersourcing [120], where users voluntarily make contributions while learning in their free time. Several paid crowdsourcing studies motivate continued contributions by providing a user with a personal recommendation to learn something new about themselves [222]. The "Did you know" automatic insights from Chapter 7 could also be perceived as a type of recommendation post contribution.

The Task Requires you to:

1. Collaborate with other people to complete the task
2. Complete the task with Artificial Intelligence
3. Complete the task by yourself
4. Learn something new
5. Contribute or use specialized knowledge you already know
6. Provide your personal information

Chapters 4, 5, 6, and 7 often require participants to contribute specialized knowledge they know in the form of sketches or interpretations of Computer Science research areas. Building on this idea, prior research on knowledge-intensive crowd contribution tasks [61] shows that paid crowdworkers [86, 169] can provide accurate contributions requiring domain-specific knowledge when they also have the required knowledge. Prior research creating discrete choice experiments also found a relationship between expertise and stated preference when selecting healthcare treatments [52]. It is also common for requesters to require paid crowdworkers to learn new skills or knowledge to make accurate contributions for a paid micro-task [61, 241], such as those covered in chapters 4 and 5.

Drafty and Sketchy allow their users to complete the task independently, whereas recent research studies scenarios where people and AI collaborate on tasks [235]. Chapter 5 explores collaborative editing behaviors among paid crowdworkers. Often, peer production tasks produce a sense of collaboration among contributors [16]. Many of these prior research efforts, show it is common for paid crowdworkers to learn something new to contribute [123]. Even unpaid citizen science efforts often require users to gain new knowledge to contribute [34].

Many surveys require a participant to provide their personal information. This is often used to improve the validity and generalization of results. However, specific paid crowdsourcing platforms like Prolific have begun providing this information automatically to potentially help protect worker privacy and reduce task completion time.

Who Asks you to Complete the Task:

1. A friend

2. A family member

3. Someone you do not know

4. People on social media

5. A bot (not a person) on social media

6. A for-profit company

7. A non-profit company

8. A system (i.e., Wikipedia or Drafty)

9. Paid Crowdsourcing system (i.e., Prolific)

While paid crowdsourcing platforms such as Prolific or Mechanical Turk are one of the most common places to find crowd contribution tasks, there are other platforms and people asking people to contribute. For example, in Chapters 3 and 5, Drafty asks its everyday users to make crowd contributions. Moving beyond systems, prior research into "friendsourcing" shows asking your friends to contribute increases the quality and likelihood of contributions [28, 29]. If a friend is not asking you to contribute, a likely alternative is a family or someone you do not know. Comparing these three options will help define how vital an individual's network is when attracting people to make crowd contributions. Research also studies when bots on Twitter ask people to contribute [200]. Beyond friendsourcing, Rogstadius et al. studied the quality and likelihood of crowd contributions of paid crowdworkers when posting the tasks as either for-profit or non-profit companies [192]. Thus this attribute integrates this aspect when comparing with other possible entities or people that can ask for crowd contributions.

What Happens with your Contribution?

1. You own the data you contributed (you can see and edit it)

2. You do not own the data you contribute (you cannot see or edit it)

3. A public community owns the data you contributed (anyone can see and edit it)

4. You receive no credit for your contribution (it is anonymous)

5. Your name or username is attached to your contribution (not anonymous)

6. Your contribution is automatically accepted

7. Your contribution could be rejected

Prior research shows that a contribution being rejected can de-motivate people [150]. Even new Wikipedia users can be dissuaded by the possibility of another user quickly reverting or rejecting their edit [181]. Drafty employs the model that someone's contribution is automatically accepted. Unlike Wikipedia and WikiData [96], Drafty and Sketchy choose to make contributions anonymous by default. Originally the intention was to reduce the effort required to contribute. Even sites similar to StackOverflow require users to create accounts to possibly identify themselves. Also, many paid crowdsourcing systems like Prolific and Amazon Mechanical Turk to provide paid crowdworkers anonymous IDs to safeguard their personal information.

It is common in paid crowdsourcing scenarios for someone to complete a task and not have access to see or edit their contribution [122]. Imagine a paid crowdworker labeling hundreds of images but having no ability to correct a mistake. Systems like Drafty, Wikipedia, and WikiData allow users and the entire community of interested users to see or edit any contribution. This idea of community ownership of the data is a hallmark of peer production [16, 231]. Chapter 6 presents Sketchy an individual creativity support tools, where only the original contributor can edit their contributions. Surveys often allow users to see and edit their submissions; in this scenario, people own the data they contributed.

Your Perception of the Task:

1. The task looks interesting
2. The task looks boring
3. The task might be unethical
4. The task is likely ethical
5. Your contribution might help people you do not know
6. Your contribution might help your peers or community

Someone's perception of a crowd contribution task covers multiple possible levels, many of which are intrinsic motivators. Prior research shows that perceiving a task as boring is one of the main reasons paid and unpaid workers quit a task [150]. While Chapters 3, 5, and 7 show user interest increases the quality and likelihood of someone contributing. Likewise, Clauset et al. show that paid crowdworkers are more likely to contribute if the task is relevant to their interests [51]. Additional research shows older adults prefer tasks matching their interests [30]. Among paid crowdworkers, prior work shows helping others can intrinsically motivate them to contribute their time even when paying less money [192]. Beyond interest, one of the best motivators to appeal to a user's intrinsic motivation is showing how their contribution will help others [160, 200]. Will helping others in our outside their communities affect their motivation differently? Also, while many researchers use paid crowdworkers to complete tasks, there are lingering questions about how the ethical use of the data collected by paid crowdworkers could affect their motivation to contribute [80].

8.2.2 Creating the Choice Sets from the Attributes and Levels:

We use the Qualtrics built-in survey builder to create and conduct the discrete choice experiment. Qualtrics automatically constructs the choice sets from the provided attributes and levels. Participants selected their preferred crowd contribution task a total of twenty times.

Qualtrics utilizes a randomized balanced design approach to ensure the choice sets are varied and present all levels to each participant. This approach combines well with Hierarchical Bayesian estimation techniques, which Qualtrics uses to analyze participant's choice data.

The design approach consists of choice sets, see Figure 8.1. Each choice set contains the same number of attributes and one level per attribute. The algorithm generates random bundles for each attribute and level, then checks each choice set to ensure relative balance in the number of times someone sees each level. Qualtrics' algorithm does not force each level to appear the same number of times per participant.

Qualtrics design does ensure the difference between the level seen the most per participant and the level seen the least is no more than a deviation of two. If Qualtrics fails to generate twenty choice sets that do not meet these criteria, it regenerates them again until the criteria are satisfied.

Recruiting Participants

The goal of the main study is to recruit normal users of Drafty to take the survey. This includes both paid crowdworkers and unpaid contributors to Drafty. To accomplish this, Drafty uses a banner; see Figure 8.2.

8.2.3 Study Procedures

Crowd Contribution Tasks within Drafty

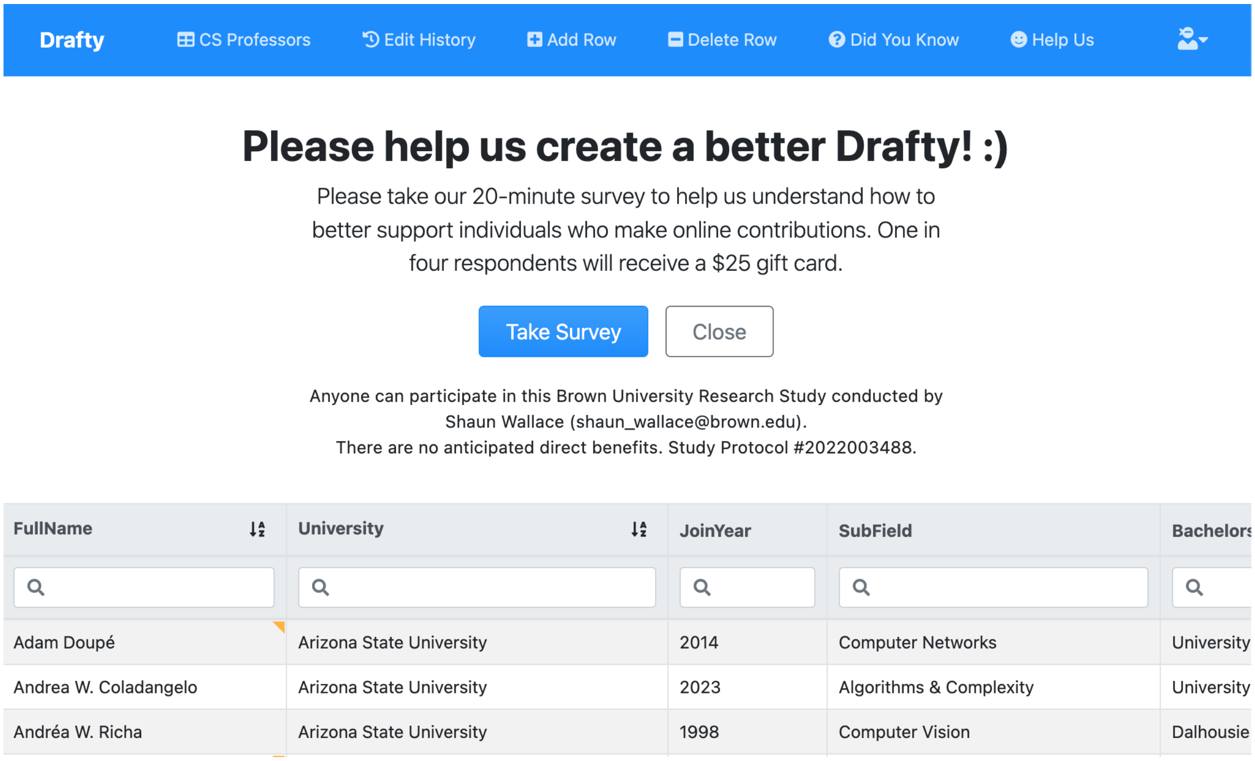
This study features paid crowdworkers and unpaid contributors editing Computer Science professor data within Drafty. They add, update, or delete data in Drafty by completing one of six possible tasks:

1. Fill in an empty cell within Drafty (fix 1 empty cell)

2. Add a new row of data (add a new Professor)

3. Delete a row of existing data (remove a Professor)

4. Review a row of existing data (review and fix a Professor's data)



Please help us create a better Drafty! :)

Please take our 20-minute survey to help us understand how to better support individuals who make online contributions. One in four respondents will receive a $25 gift card.

Anyone can participate in this Brown University Research Study conducted by Shaun Wallace (shaun_wallace@brown.edu).
There are no anticipated direct benefits. Study Protocol #2022003488.

FullName	University	JoinYear	SubField	Bachelors
Adam Doupé	Arizona State University	2014	Computer Networks	University
Andrea W. Coladangelo	Arizona State University	2023	Algorithms & Complexity	University
Andréa W. Richa	Arizona State University	1998	Computer Vision	Dalhousie

Figure 8.2: Anyone can voluntarily choose to take the survey and will have a one in four chance to receive a $25 Amazon gift card as compensation. To be eligible, participants must enter their email in a separate survey that is provided after and not connected to the main discrete choice experiment survey.

5. Add a note to a row (add a note about a Professor)

6. Contributed using the "Help Us!" feature within Drafty

While paid crowdworkers are free to perform any task within Drafty, they are instructed to "Add a new row of data." This provides the opportunity for each paid crowdworker to find and add six pieces of information for a Computer Science professor not currently listed in Drafty.

Recruiting Drafty's Visitors: Unpaid Contributors and Paid Crowdworkers

While discrete choice experiments have good internal validity, we need observable data to improve the result's external validity. To enhance external validity, everyday visitors from Drafty can choose to take the discrete choice experiment survey. These everyday visitors will be Drafty's regular unpaid contributors and paid crowdworkers from Prolific. During the recruitment period, Drafty contains a banner to recruit participants. Anyone who completes the survey has a one-in-four chance to win a $25 Amazon Gift Card.

Unpaid contributors are recruited using the sourcing methods from Chapter 7. These include organic visitors (i.e., search engines), visitors from dynamic sources like CS Open Rankings or the Did You Know Twitter account, asking people to visit Drafty to review data using email or Twitter, and making posts on public forums such as Reddit to advertise Drafty's dataset of computer science professors.

Paid crowdworkers are recruited using Prolific, an online research platform to recruit paid crowdworkers by posting micro-tasks. We used Prolific's built-in pre-screening to ensure paid crowdworkers met the following criteria:

1. Minimum 95% approval rate.
2. Minimum 100 tasks completed.
3. Minimum age of 18.
4. They could not have completed any our prior tasks, to ensure new unique participants each time.
5. Are from the USA. (Because Drafty mainly features universities from the US and Canada.)
6. Are using a Desktop device, because the user interface design of Drafty the survey are optimized for Desktop.

All tasks we advertised as a 15 minute estimated completion time based on timings from our pilot study. When creating micro-tasks, participants were compensated within Prolific payment limits and guidelines at $8.00, $12.00, or $16.00 per hour. Prolific's default recommendation is $12.00 per hour. At the time of this study, the maximum Prolific's interface reports to requesters are $16.00 per hour, and the minimum is $8.00 per hour. These intervals align with Prolific's recommendations to requesters when assigning monetary compensation to a micro-task. These intervals also align with the levels the discrete choice experiment uses for payment per hour. Thus, this compensation scheme aligns with real-world examples of recruiting paid crowdworkers across Prolific.

Each paid crowdsourcing task asked paid crowdworkers to review Drafty's data and add a missing professor from a university. Table 8.2 shows the university name and payment per task. The universities for the paid crowdsourcing task were selected because they had either not had a recently added professor recently, or they are new options to add professors to Drafty (i.e., Drafty has no data for this university.). Also, we hand-checked each university for missing professors and new assistant professors to ensure there were enough professors to add following a similar study and recommendations from Papoutsaki et al. [173].

For each university posted on Prolific, we emailed the department chairs and at least one other faculty member from the university's Computer Science department. This provides a chance for paid crowdworkers and unpaid contributors alike to review and add data. We also emailed department chairs in Chapter 7 as part of the "asking" recruitment method.

Prolific was used to post all paid crowdsourcing tasks. An example title used on the posts was "Help Build a Dataset of Computer Science Professors - University of Arizona." The instructions posted on Prolific for the paid crowdworkers include the following messages:

University	Pay Level per hour
Case Western Reserve University	$8
George Washington University	$16
Georgetown University	$12
Illinois Institute of Technology	$12
Rensselaer Polytechnic Institute	$12
Temple University	$16
University of Arizona	$8
University of California, Davis	$16
University of Delaware	$8
University of Florida	$12
University of Houston	$16
University of Maryland, Baltimore County	$8
University of Oregon	$12
University of South Florida	$8
University of Tulsa	$16

Table 8.2: The universities used to create tasks on Prolific. There are five universities per pay level. The new universities added to Drafty as part of the study were the Illinois Institute of Technology, Temple University, and the University of South Florida. These were added to provide paid crowdworkers and unpaid contributors with around 25 new professors to add per university. The other universities listed were universities with the longest duration since a new professor was added in Drafty.

INITIAL MESSAGE You are invited to take part in a Brown University research study. Your participation is voluntary. :)

PURPOSE This study focuses on collecting information about specific computer science faculty members. Drafty is a public data system with thousands of computer science faculty profiles from the US and Canada.

PROCEDURES You have to add one new professor from the [UNIVERSITY NAME], not currently listed in Drafty. One row consists of a professor's:

1. Full Name
2. University (where they work at)
3. Join Year (the year they started as a professor at that university)
4. SubField (their primary research area)
5. Bachelors (the university where they got their bachelor's degree)
6. Doctorate (the university where they got their PhD)

Steps

1. Open this link ([URL to faculty webpage for the university]) to visit this faculty listing page. Keep this page open.
2. Visit Drafty using Prolific's study link (Open the study link in a new window)

3. Compare the professors listed on Drafty with the webpage from step 1.

4. Add one missing professor to Drafty. This is a professor listed on their webpage but not on Drafty.

5. To add a new row, please select the white text ”Add Row” in the blue bar at the top of the page. After adding a new row, you see a modal pop-up window with the Prolific completion code at the bottom.

*They must be a tenure-track professor who can solely advise a CS PhD student. They should have the title Assistant Professor, Associate Professor, or Full Professor.

COMPENSATION You will receive a base payment through Prolific for adding one new row of data to Drafty. You must add a new professor.

By recruiting Drafty’s everyday visitors, we can compare their stated preferences for contributing to tasks with their real-world behavioral data collected within Drafty. For example, did they contribute data, or were they lurking? Thus, providing evidence of what motivates people who contribute to an existing public data platform such as Drafty.

8.3 Results

This research asks the following questions: 1) what parameters to design fair and equitable incentive mechanisms for crowd contributions apply to most users of a public data system, and 2) what parameters should be individuated per group of users?

The following data and results sections will answer these questions by reviewing responses to a survey-based study taken by real-world users of Drafty.

8.3.1 Data Overview

The study was conducted from March 11th to April 25th, 2023. During this time, 149 people freely chose to begin the survey after visiting Drafty. These people consist of paid crowdworkers and unpaid contributors (i.e., Drafty’s normal everyday visitors). During the study time, 115 paid crowdworkers were recruited to add new rows of data to Drafty. During the same time, there were 2,723 visits to Drafty’s spreadsheet interface. These everyday active visitors freely chose to visit Drafty and made at least one interaction with Drafty’s spreadsheet interface (i.e., click, search, edit, etc.). All visitors to Drafty made 33,283 total interactions and 1,048 edits during this study timeframe in 2023.

Survey Data Pre-Processing

The following section describes the survey study data’s review process and removal criteria. A total of 149 people chose to take the survey. Based on the session and profile ids tracked from Drafty within the survey, the same participant never took the survey multiple times. Among the initial 149 participants, 16 were removed for not completing the survey. After that, 10 additional participants were removed for entering the same response for ten Likert-scale questions. Finally, 2 additional participants were removed for providing non-sensical answers to qualitative questions indicating they might be a bot or speeding through the survey. After answering a sample question, the remaining 122 participants reported understanding the instructions for the discrete choice experiment. The following metrics are computed using the remaining 121 participants. The average completion time is 19 minutes and 6 seconds. The median completion time for the survey is 16 minutes and 11 seconds. All participants completed the survey within two standard deviations of the median completion time; hence no one completed the survey abnormally fast.

Survey Participants

The following summary demographics are for the remaining 121 participants who completed the survey. The average age per participant was 36 years, ranging from 19 to 78. Among several options to indicate gender identity, 58 participants identified as Male and 43 as Female, while 20 preferred not to say or chose another option. Among all participants, 67 indicated payment for a task is for extra spending money, while 26 indicated it helps pay some of their bills.

8.3.2 Evaluation Metrics

The metrics described below are used throughout this chapter's results section. The values for these metrics are computed using Qualtrics. These are common analyses used to evaluate discrete choice experiments [170]. Within these metrics, the terms utility and preference relate to someone's motivation for completing a crowd contribution task.

Average Utility Scores are the average utility score of each level across all participants. The utility scores show the relative preference between levels within an attribute. A level with a positive utility score will increase someone's motivation to complete a crowd contribution task, while a negative score will decrease it.

Preference Share measures the probability that a level would be chosen over another when all other attribute levels are the same. It is computed using a Multinomial Logistic Regression model and the utility scores per level within an attribute. In section 8.3.5, the analysis uses Preference Share to simulate the total utility (i.e., preference) between two different crowd contribution tasks.

Attribute Importance is an attribute's influence on someone's preference for a crowd contribution task. The greater the attribute importance, the more influence its levels have over someone's preference. Attribute importance is computed by taking the difference between the average utility scores from the best and worst levels within that attribute. The larger this difference, the more important the attribute. If an attribute has a highly desirable or undesirable level, this can influence the attribute's importance.

Optimal Task describes the optimal crowd contribution task. It is the combination of the one level per attribute with the most influence on someone's preference or motivation. Normally, this is the level per attribute with the highest utility score.

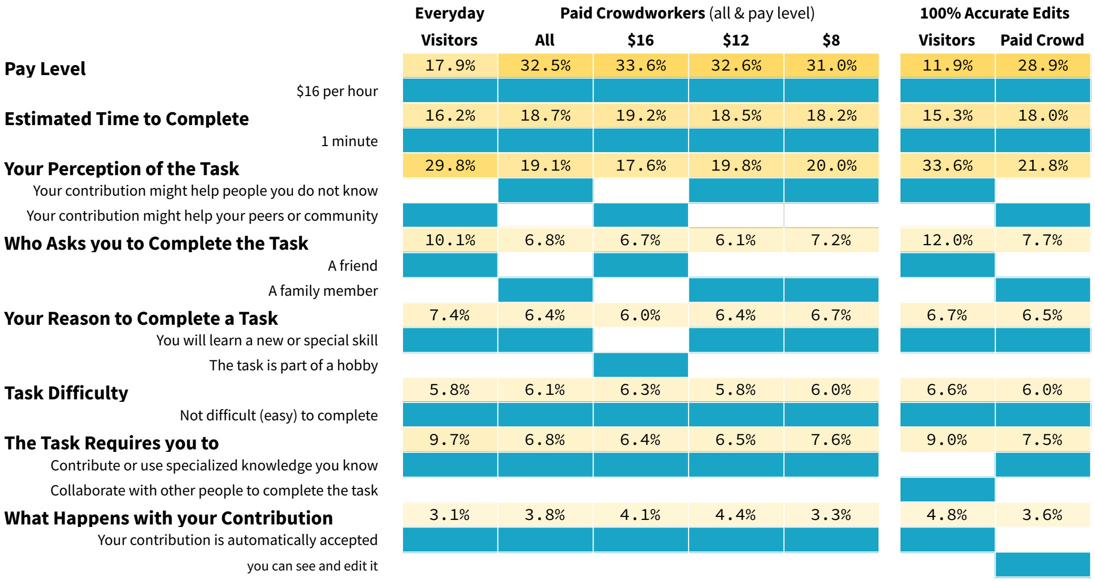
Trade-offs between two attributes and their levels can be computed using partworth functions [182]. These results can help inform if one level is changed within attribute "A", and how the level of attribute "B" need to increase or decrease to produce the same utility (i.e., level o motivation). For example, if the estimated time to complete a task is increased, how much should the pay level increase to maintain the same level of motivation? To compute a partworth function, an attribute's level, and utility score must be known.

8.3.3 Universal Incentives are Pay Level, Time to Complete, and Task Perception

If you are creating crowd contribution tasks, focus on improving the pay level, reducing the estimated time to complete, and improving someone's perception of the task (i.e., showing how the contribution can help others and make it interesting). Figure 8.3 shows unpaid contributors and paid crowdworkers share similar motivations about the most important level per attribute. The most important attributes and levels are universal across users who completed the survey. However, when scrutinizing the results further everyday users and paid crowdworkers make different trade-offs when selecting between crowd contribution tasks.

While unpaid contributors and paid crowdworkers share similar motivations, task perception was most important to unpaid contributors, and pay level was most important to paid crowdworkers. For paid crowdworkers, there was a 74% increase in the attribute importance for pay level compared to the estimated time to complete. At the same time, there was only a 10% increase among unpaid contributors. This finding that paid crowdworkers value pay level the most mirrors prior research [152]. Also, when comparing attribute importance for pay level and task perception, there is a 70% increase among paid crowdworkers for pay level. At the same time, there is a 40% decrease among unpaid contributors for pay level. While the pay level motivates unpaid contributors to complete crowd contribution tasks, task perception plays a larger role. The levels for task perception mainly focus on intrinsic motivators, thus showing unpaid contributors are influenced by intrinsic motivators [159, 224].

When comparing feature importance for those who only made 100% accurate edits on Drafty, unpaid contributors and paid crowdworkers were less influenced by the pay level per task compared to people from the same population who completed the survey. Also, both unpaid contributors and paid crowdworkers were more influenced by task perception. Indicating someone's level of interest and intrinsic motivators can play a more influential role in their motivation than money. This result mirrors prior research on paid crowdworkers [51] and demonstrates a similar result among unpaid contributors. Notably, the feature importance between pay level and who is asking for a contribution is similar among unpaid contributors who made 100% accurate edits on Drafty. This result mirrors prior research by Brady et al. [28]



	Everyday	Paid Crowdworkers (all & pay level)				100% Accurate Edits	
	Visitors	All	$16	$12	$8	Visitors	Paid Crowd
Pay Level	17.9%	32.5%	33.6%	32.6%	31.0%	11.9%	28.9%
$16 per hour	■	■	■	■	■	■	■
Estimated Time to Complete	16.2%	18.7%	19.2%	18.5%	18.2%	15.3%	18.0%
1 minute	■	■	■	■	■	■	■
Your Perception of the Task	29.8%	19.1%	17.6%	19.8%	20.0%	33.6%	21.8%
Your contribution might help people you do not know		■		■	■	■	
Your contribution might help your peers or community	■		■				■
Who Asks you to Complete the Task	10.1%	6.8%	6.7%	6.1%	7.2%	12.0%	7.7%
A friend	■		■			■	
A family member		■		■	■		■
Your Reason to Complete a Task	7.4%	6.4%	6.0%	6.4%	6.7%	6.7%	6.5%
You will learn a new or special skill	■	■		■	■	■	■
The task is part of a hobby			■				
Task Difficulty	5.8%	6.1%	6.3%	5.8%	6.0%	6.6%	6.0%
Not difficult (easy) to complete	■	■	■	■	■	■	■
The Task Requires you to	9.7%	6.8%	6.4%	6.5%	7.6%	9.0%	7.5%
Contribute or use specialized knowledge you know	■	■	■	■	■		■
Collaborate with other people to complete the task						■	
What Happens with your Contribution	3.1%	3.8%	4.1%	4.4%	3.3%	4.8%	3.6%
Your contribution is automatically accepted	■	■	■	■	■	■	
you can see and edit it							■

Figure 8.3: Unsurprisingly, all participants preferred the highest pay level per hour, the lowest estimated time to complete, and the easiest tasks. While task perception was the most important attribute for unpaid contributors, paid crowdworkers were mainly influenced by the pay level per hour. This figure uses a teal square to indicate the level per attribute with the highest average utility per attribute per group. Each column then constructs the optimal or ideal crowd contribution task per group of people. The percentages with the gold background are the relative importance measure [182] per attribute. These numbers are the normalized values from the feature importance per attribute per group. For example, the feature importance for pay level for unpaid contributors with 100% accuracy is 11.6, and the sum of the feature importance per attribute for this group is 97.2. Thus, 11.6 divided by 97.2 is 11.9%.

and Chapter 7 showing that asking friends to contribute can motivate accurate contributions.

The analysis also splits participants (paid crowdworkers and unpaid contributors) by their self-reported knowledge of Computer Science and their self-reported interest in Computer Science. Participants who self-reported High or Very High were placed in one group, and then the other group consisted of everyone else. Their optimal task design mirrored those in Figure 8.3. One notable difference is unpaid contributors with high levels of knowledge or interest in Computer Science were more influenced by tasks that required them to collaborate with people. This observation mirrors the long-term success of public systems like Drafty, like other peer production-inspired systems, where people are motivated by the idea of collaborating with others to complete tasks [16, 224].

8.3.4 Paid Crowdworkers and Unpaid Contributors Share Similar Motivations but Make Different Trade-offs When Selecting Tasks

This section compares the trends in relative utility scores between unpaid contributors and paid crowdworkers across both groups in total and among those users who made edits on Drafty and only made 100% accurate edits. The levels of each attribute show patterns about how each group differs. We created figures showing heatmaps of the relative utility scores per group; see Figures 8.4, 8.6, 8.5, and 8.7.

This analysis compares groups where one group (the ideal group) is likely to benefit Drafty more than the other (less ideal). Oftentimes, the ideal group edits data or only makes 100% accurate edits. Figures 8.5 and 8.7 show pairs of these groups separated by a column of white space, where the group on the left is ideal while the group on the right is less ideal. For example, an ideal group is unpaid contributors who only made 100% edits versus those who did not. Or, paid crowdworkers who submitted edits to Drafty versus those who abandoned the task (but still completed the survey). Lastly, paid crowdworkers who only made 100% edits versus those who submitted edits and were not 100% accurate.

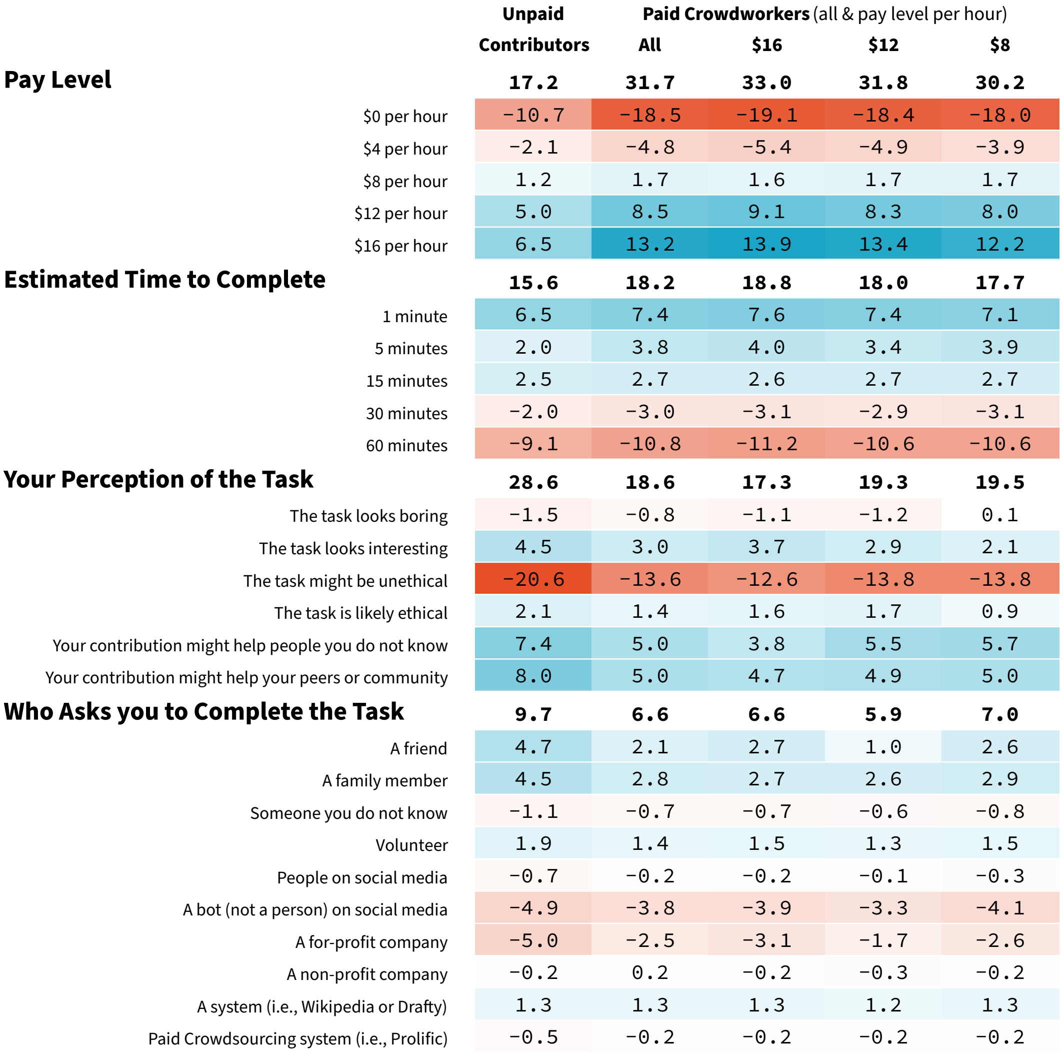
Pay Level Per Hour Across all groups in Figures 8.4 and 8.5, a minimum pay level generated a positive relative utility score. In other words, Prolific's fair payment guidelines apply even among unpaid contributors. Where preferences shift is among the highest pay level per hour. The relative utility score for the highest pay level per hour ($16) was 2 times higher for paid crowdworkers compared to unpaid contributors, see Figure 8.4. This trend continues when splitting users based on their real-world usage of Drafty (i.e., did they make an edit, were they accurate). The relative utility scores for pay level per hour ($16) for unpaid contributors who did not make edits was 2.2 times higher than those who only made accurate edits. Likewise, The relative utility scores for pay level per hour ($0) for unpaid contributors who did not make edits was 1.7 times lower than those who only made accurate edits. The ideal unpaid contributors (i.e,. those who only make accurate edits) valued other aspects of a task far more than pay level. These exact trends continue for $16 and $0 per hour when comparing the paid crowdworkers who made 100% accurate edits on Drafty. This indicates that accurate users are motivated by more than money [160]. Thus, when posting paid crowdsourcing tasks that pay more, this might attract a different type of crowdworker.

Estimated Time to Complete For estimated time to complete all groups preferred tasks under 15 minutes. While Figure 8.4 shows a trend where less time yields more preference among paid crowdworkers, this same trend does not exist among unpaid contributors. For unpaid contributors, the relative utility score for a 15 minute task is .5 points higher than for 5 minute task. This result replicates prior research showing unpaid contributors will spend more time on a task than paid crowdworkers [112]. Paid crowdworkers who submitted edits to Drafty had the same relative utility score for 15 minutes tasks and 5 minute tasks. The paid crowdworkers who did not submit edits were the ones who chose to complete the survey for a possible gift card but abandoned the initial editing task. Viewing their survey responses they felt it was difficult to contribute and find the professor's information. Since their preferences indicate they prefer shorter tasks, it is not a surprise they abandoned the task to edit data on Drafty. There was no difference in trends among paid crowdworkers who submitted 100% accurate edits and those who did not.

Your Perception of the Task There are multiple trends among the levels across groups for task perception. In every comparison, the ideal group prefers tasks that are ethical, look interesting, and will help others compared to another group. Hence, this is why the attribute importance for task perception is higher among the ideal groups. While we never condone creating tasks or gaining contributions for unethical reasons,unpaid contributors who only made accurate edits valued ethical tasks that help others more than any other group. Also, building on prior research, their interest in the task is more motivating than any other group [51, 225]. Peer production environments rely on people freely engaging and contributing data [16]. These results show how systems like Drafty motivate accurate contributions.

Who Asks you to Complete the Task We often ask others to complete crowd contribution tasks such as surveys, or at times we even volunteer to edit Wikipedia. During this dissertation, we have asked people we did not know, friends, or even had Drafty ask people to contribute. Across all groups, people preferred tasks where either they volunteered or a friend, family member, or a system (i.e., Wikipedia or Drafty) asked them to contribute. While our survey results show unpaid contributors to Drafty were the most influenced by tasks where they could volunteer, it was ultimately family and friends that matter most to all groups. This is not surprising, as Brady et al. showed that when a friend asks, this can elicit accurate contributions [29]. The relative utility score for when a friend asks is 2.2 times greater for unpaid contributors compared to paid crowdworkers and 2 times greater for unpaid contributors who only submitted accurate edits compared to those who did not. Mirroring other prior research, all groups showed a strong dislike for tasks where a bot (not a person) on social media [200] or a profit for-profit company [192] asks them to contribute. While Rogstadius et al. showed that people prefer non-profit companies compared to for-profit companies, our results across groups indicate this alone is not enough to motivate someone to complete a task and is relatively similar to people on social media asking for contributions. Maybe, to increase motivation, non-profit companies can show how people completing crowd contribution tasks for them can help others.

Your Reason to Complete a Task All groups preferred tasks where they will learn a new or special skill or it is part of a hobby. These results mirror those from learnersourcing, where people simultaneously learn new skills relate to an area of interest or a hobby, and then make contributions back to the system [120]. Notably, in this survey-based study receiving reputation points in a system (badges, points, credit, etc.) produced a consistently negative effect across all groups. While learning communities have successfully used gamification to increase engagement [157], Drafty's users (paid and unpaid) did not show a preference for this motivator. This result mirrors prior research shows how



	Unpaid	Paid Crowdworkers (all & pay level per hour)			
	Contributors	All	$16	$12	$8
Pay Level	**17.2**	**31.7**	**33.0**	**31.8**	**30.2**
$0 per hour	-10.7	-18.5	-19.1	-18.4	-18.0
$4 per hour	-2.1	-4.8	-5.4	-4.9	-3.9
$8 per hour	1.2	1.7	1.6	1.7	1.7
$12 per hour	5.0	8.5	9.1	8.3	8.0
$16 per hour	6.5	13.2	13.9	13.4	12.2
Estimated Time to Complete	**15.6**	**18.2**	**18.8**	**18.0**	**17.7**
1 minute	6.5	7.4	7.6	7.4	7.1
5 minutes	2.0	3.8	4.0	3.4	3.9
15 minutes	2.5	2.7	2.6	2.7	2.7
30 minutes	-2.0	-3.0	-3.1	-2.9	-3.1
60 minutes	-9.1	-10.8	-11.2	-10.6	-10.6
Your Perception of the Task	**28.6**	**18.6**	**17.3**	**19.3**	**19.5**
The task looks boring	-1.5	-0.8	-1.1	-1.2	0.1
The task looks interesting	4.5	3.0	3.7	2.9	2.1
The task might be unethical	-20.6	-13.6	-12.6	-13.8	-13.8
The task is likely ethical	2.1	1.4	1.6	1.7	0.9
Your contribution might help people you do not know	7.4	5.0	3.8	5.5	5.7
Your contribution might help your peers or community	8.0	5.0	4.7	4.9	5.0
Who Asks you to Complete the Task	**9.7**	**6.6**	**6.6**	**5.9**	**7.0**
A friend	4.7	2.1	2.7	1.0	2.6
A family member	4.5	2.8	2.7	2.6	2.9
Someone you do not know	-1.1	-0.7	-0.7	-0.6	-0.8
Volunteer	1.9	1.4	1.5	1.3	1.5
People on social media	-0.7	-0.2	-0.2	-0.1	-0.3
A bot (not a person) on social media	-4.9	-3.8	-3.9	-3.3	-4.1
A for-profit company	-5.0	-2.5	-3.1	-1.7	-2.6
A non-profit company	-0.2	0.2	-0.2	-0.3	-0.2
A system (i.e., Wikipedia or Drafty)	1.3	1.3	1.3	1.2	1.3
Paid Crowdsourcing system (i.e., Prolific)	-0.5	-0.2	-0.2	-0.2	-0.2

Figure 8.4: Attribute importance per attribute is the bold number in the same row as the attribute name. This is the difference between the level with the highest and lowest relative utility level per attribute.

gamification mechanisms can decrease the quantity and quality of crowd contributions over time [147]. Considering how strong of an intrinsic motivator helping others is, the extrinsic motivator of earning badges does not matter in Drafty's context. Unpaid contributors who only submitted accurate edits were almost twice as motivated by tasks where they might be paid for doing exceptional work. While Drafty does not mix an extrinsic motivator such as money for unpaid contributors, this could be a potential future direction that has produced accurate contributions in the past [31]. This idea of possibly paying others for doing exceptional work produced a relative utility score that is 181% greater among unpaid contributors who only submitted accurate edits compared to those who did not. This is additional evidence that mixing extrinsic rewards in Drafty could further motivate its existing highly accurate visitors.

	Unpaid Contributors		Paid Crowdworkers		Paid Crowdworkers	
	Acc. Edits	No Edits	Edits	No Edits	100% Acc.	<100% Acc.
Pay Level	**11.6**	**19.6**	**27.9**	**33.7**	**28.1**	**31.9**
$0 per hour	-7.0	-12.0	-15.5	-20.2	-16.6	-18.7
$4 per hour	-2.6	-1.9	-5.5	-4.3	-4.0	-4.7
$8 per hour	1.5	1.1	1.4	1.8	1.5	1.7
$12 per hour	4.6	5.2	7.2	9.2	7.6	8.4
$16 per hour	3.5	7.6	12.4	13.5	11.5	13.2
Estimated Time to Complete	**15.0**	**15.9**	**17.1**	**18.7**	**17.5**	**18.2**
1 minute	5.9	6.9	6.5	7.8	7.2	7.4
5 minutes	2.5	1.7	3.3	4.0	3.2	3.8
15 minutes	2.3	2.6	3.3	2.4	2.6	2.7
30 minutes	-1.6	-2.2	-2.6	-3.3	-2.8	-3.1
60 minutes	-9.1	-9.0	-10.6	-10.9	-10.3	-10.8
Your Perception of the Task	**32.9**	**26.7**	**21.3**	**17.3**	**21.2**	**18.5**
The task looks boring	-2.2	-1.1	-0.3	-1.0	-0.9	-0.7
The task looks interesting	5.2	4.3	2.9	3.0	3.5	3.0
The task might be unethical	-23.6	-19.3	-15.5	-12.6	-15.5	-13.5
The task is likely ethical	2.9	1.8	1.7	1.3	1.6	1.4
Your contribution might help people you do not know	8.4	6.9	5.8	4.7	5.6	4.8
Your contribution might help your peers or community	9.3	7.4	5.5	4.6	5.7	5.0
Who Asks you to Complete the Task	**11.7**	**8.9**	**7.3**	**6.1**	**7.5**	**6.5**
A friend	6.3	3.1	2.8	1.8	2.8	2.4
A family member	3.5	4.3	3.3	2.5	3.3	2.7
Someone you do not know	-0.8	-1.1	-0.6	-0.7	-0.8	-0.7
Volunteer	1.5	1.8	1.6	1.3	1.6	1.4
People on social media	-1.2	-0.4	-0.5	0.0	-0.3	-0.2
A bot (not a person) on social media	-4.1	-4.6	-4.0	-3.6	-4.2	-3.8
A for-profit company	-5.4	-4.0	-3.0	-2.2	-3.2	-2.5
A non-profit company	-0.2	-0.1	-0.6	0.0	-0.2	-0.3
A system (i.e., Wikipedia or Drafty)	0.9	1.3	1.4	1.2	1.3	1.2
Paid Crowdsourcing system (i.e., Prolific)	-0.5	-0.4	-0.3	-0.2	-0.3	-0.2

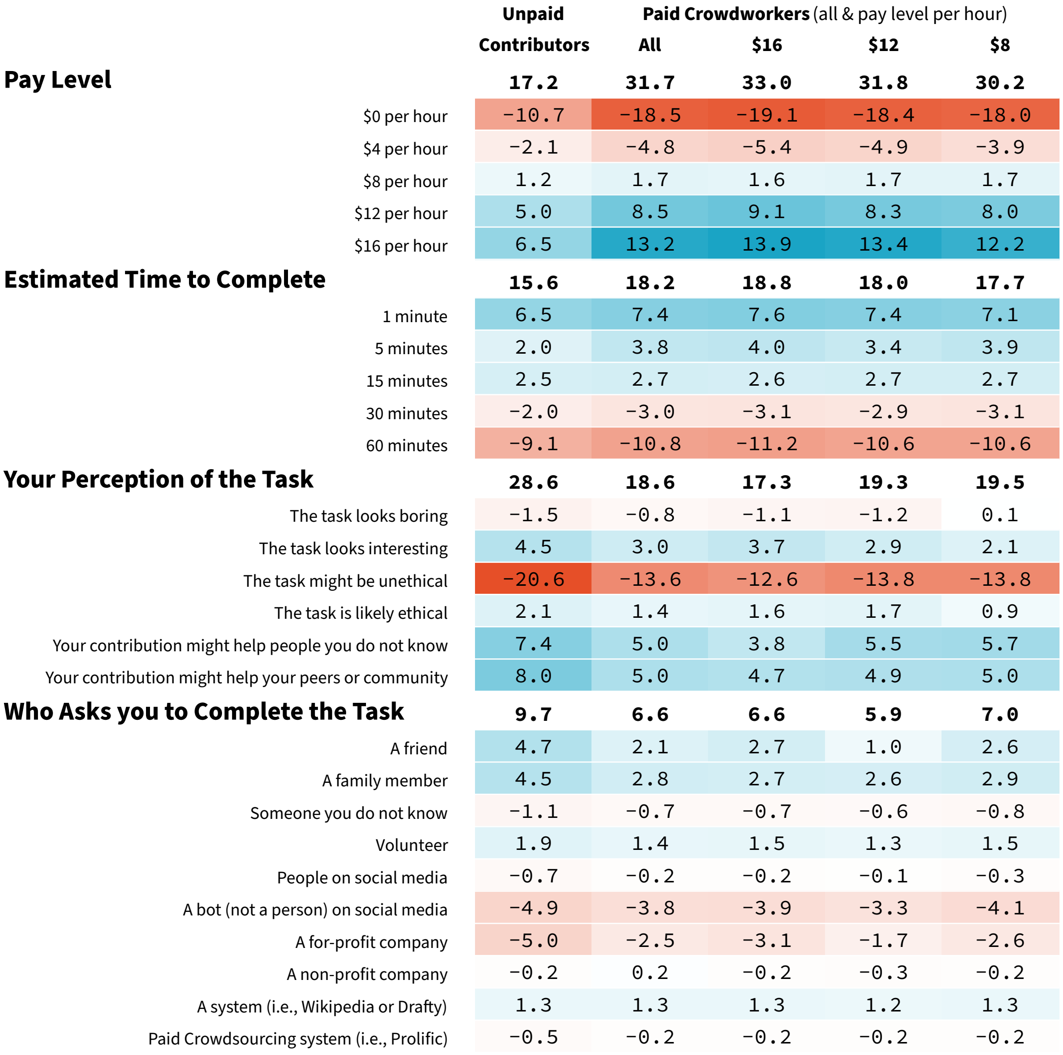
Figure 8.5: Attribute importance per attribute is the bold number in the same row as the attribute name. This is the difference between the level with the highest and lowest relative utility level per attribute.

Task Difficulty Task difficulty is one of the most consistent attributes across this study in terms of affecting the preferences and motivation of all users. While task difficulty is often optimized by system designers [101], it is also cited as affecting people's perceptions of task time [140]. It is not a surprise that crowdworkers paid $16 per hour valued easy tasks more than others. They are trying to optimize their actual earnings per hour. Among unpaid contributors, those who did not submit edits were the only group with a positive relative utility score for moderately difficult tasks. Maybe this group did not edit data on Drafty because the tasks to contribute were too simple and straightforward. Future research could assess how to appeal to users who only become contributors over time.

The Task Requires you to Drafty often requires people to contribute or use specialized knowledge you know for data types requiring domain-specific expertise. While this positively motivates every group, unpaid contributors who only made accurate edits preferred to collaborate with others the most. Perhaps collaboration can lead to a greater sense of community and contributions to helping others, which is why this group preferred collaboration the most [133]. The relative utility score for collaborating with unpaid contributors was 10 times higher than paid crowdworkers. It is possible that paid crowdworkers might view that collaborating with others could also increase the time to complete. Looking at possible future AI-centric trends in crowdsourcing, while Drafty does not integrate an AI to help people complete tasks, there were trends in participant responses. Unpaid contributors who only made accurate edits had the strongest negative preference for tasks requiring collaborating with an AI. In contrast, they had the strongest positive preference for collaborating with other people. Paid crowdworkers were open to collaborating with AI to complete a task. This mirrors a growing trend where AI is being used to recommend tasks [41].

What Happens with your Contribution In paid crowdsourcing, it is common for people to make contributions and then never have access to these again. Bernstein raised this concern when discussing his crowd-powered system Soylent [22].

All groups had a negative preference for tasks where they could not see or edit their contribution. The relative utility score for crowdworkers paid $12 an hour was 25% lower than those paid $16 per hour. The more crowdworkers were paid, the less their ownership of their contribution mattered. However, when we compare paid crowdworkers who only made accurate edits versus those who did not, the relative utility scores were similar. Building on this idea, the relative utility score for contributions where people can see and edit it was 5 times greater for paid crowdworkers who submitted edits versus those who did not. This trend continued among unpaid contributors, where the relative utility score was 1.6 for those who made accurate edits compared to 0.0 for those who did not. Ensuring contributors (paid and unpaid) can see and edit their contributions will motivate accurate contributions. Two other trends across groups are that everyone preferred tasks where they could remain anonymous, and their contribution was automatically accepted. Both of these levels mirror Drafty's system design. There is one notable exception for the preference of having a username attached to contributions (not anonymous). The relative utility score for contributions being not anonymous for everyay visitor who did not contribute is 5.6 times greater compared to those who only submitted accurate edits. Within Drafty's spreadsheet interface, credit is provided per row or cell to any contributions. Perhaps, this group of unpaid contributors who did not contribute could be motivated if there was more obvious attribution provided for their efforts.



	Unpaid Contributors	Paid Crowdworkers (all & pay level per hour) All	$16	$12	$8
Pay Level	**17.2**	**31.7**	**33.0**	**31.8**	**30.2**
$0 per hour	-10.7	-18.5	-19.1	-18.4	-18.0
$4 per hour	-2.1	-4.8	-5.4	-4.9	-3.9
$8 per hour	1.2	1.7	1.6	1.7	1.7
$12 per hour	5.0	8.5	9.1	8.3	8.0
$16 per hour	6.5	13.2	13.9	13.4	12.2
Estimated Time to Complete	**15.6**	**18.2**	**18.8**	**18.0**	**17.7**
1 minute	6.5	7.4	7.6	7.4	7.1
5 minutes	2.0	3.8	4.0	3.4	3.9
15 minutes	2.5	2.7	2.6	2.7	2.7
30 minutes	-2.0	-3.0	-3.1	-2.9	-3.1
60 minutes	-9.1	-10.8	-11.2	-10.6	-10.6
Your Perception of the Task	**28.6**	**18.6**	**17.3**	**19.3**	**19.5**
The task looks boring	-1.5	-0.8	-1.1	-1.2	0.1
The task looks interesting	4.5	3.0	3.7	2.9	2.1
The task might be unethical	-20.6	-13.6	-12.6	-13.8	-13.8
The task is likely ethical	2.1	1.4	1.6	1.7	0.9
Your contribution might help people you do not know	7.4	5.0	3.8	5.5	5.7
Your contribution might help your peers or community	8.0	5.0	4.7	4.9	5.0
Who Asks you to Complete the Task	**9.7**	**6.6**	**6.6**	**5.9**	**7.0**
A friend	4.7	2.1	2.7	1.0	2.6
A family member	4.5	2.8	2.7	2.6	2.9
Someone you do not know	-1.1	-0.7	-0.7	-0.6	-0.8
Volunteer	1.9	1.4	1.5	1.3	1.5
People on social media	-0.7	-0.2	-0.2	-0.1	-0.3
A bot (not a person) on social media	-4.9	-3.8	-3.9	-3.3	-4.1
A for-profit company	-5.0	-2.5	-3.1	-1.7	-2.6
A non-profit company	-0.2	0.2	-0.2	-0.3	-0.2
A system (i.e., Wikipedia or Drafty)	1.3	1.3	1.3	1.2	1.3
Paid Crowdsourcing system (i.e., Prolific)	-0.5	-0.2	-0.2	-0.2	-0.2

Figure 8.6: Attribute importance per attribute is the bold number in the same row as the attribute name. This is the difference between the level with the highest and lowest relative utility level per attribute.

8.3.5 Computing Trade-offs to Optimize Incentives for Paid Crowdworkers and Unpaid Contributors

The previous section identifies and discusses how groups make different trade-offs when selecting crowd contribution tasks. This section computes the trade-offs for pay level and estimated time to complete across various attributes and levels to provide insights into how future system designers and requesters can construct tasks while maintaining user motivation.

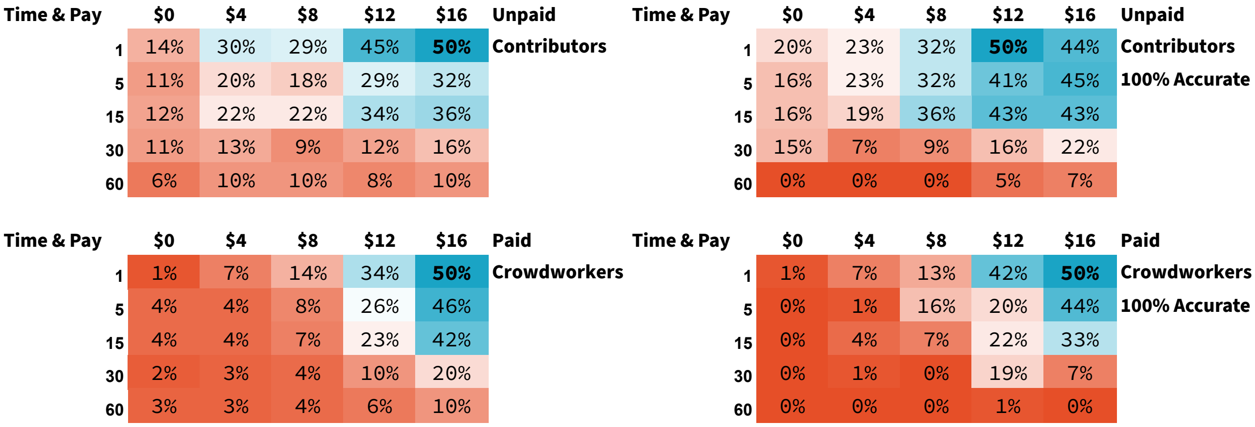
One method to view these trade-offs is comparing the preference share (i.e., total preference or utility) for a given crowd contribution task to a group's optimal crowd contribution task. Figure 8.8 shows how preferences and motivations shift as pay levels and task time are manipulated. The preference share for a task that pays $0 for unpaid contributors is 2 to 14 times greater than paid crowdworkers when comparing all estimated task completion times. This trend continues

	Unpaid Contributors		Paid Crowdworkers		Paid Crowdworkers	
	Acc. Edits	No Edits	Edits	No Edits	100% Acc.	<100% Acc.
Your Reason to Complete a Task	**6.6**	**7.4**	**6.9**	**5.8**	**6.3**	**6.1**
You might be paid for doing exceptional work	1.8	-2.2	-1.5	-1.5	-1.2	-1.8
Your contribution benefits you personally	0.4	0.9	0.8	0.4	0.6	0.6
You will learn a new or special skill	2.9	4.2	3.6	3.1	3.4	3.3
Personal rec. or learn something new about you	-0.8	0.1	0.0	0.0	-0.1	0.1
Reputation points in a system (badges, points, credit)	-3.7	-3.2	-3.3	-2.7	-2.9	-2.8
The task is part of your job	-1.5	-1.7	-2.0	-1.8	-1.8	-1.9
The task is part of a hobby	1.0	2.0	2.5	2.4	2.1	2.5
Task Difficulty	**6.5**	**5.1**	**6.1**	**5.8**	**5.8**	**6.0**
Not difficult (easy) to complete	3.3	2.0	3.1	2.9	2.8	3.0
Moderately difficult to complete	-0.1	1.1	-0.1	0.0	0.2	0.0
Very difficult to complete	-3.2	-3.1	-3.0	-2.9	-3.0	-3.0
The Task Requires you to	**8.8**	**9.5**	**7.1**	**6.4**	**7.3**	**6.6**
Collaborate with other people to complete the task	4.0	2.4	-0.8	-0.2	0.2	-0.5
Complete the task with Artificial Intelligence	-2.4	-0.1	0.5	-0.1	-0.2	0.1
Complete the task by yourself	-0.7	-0.4	-0.3	-0.3	-0.3	-0.3
Learn something new	1.0	1.1	1.7	1.5	1.5	1.6
Contribute or use specialized knowledge you know	2.8	3.3	3.0	2.8	3.0	2.8
Provide your personal information	-4.8	-6.2	-4.1	-3.6	-4.3	-3.8
What Happens with your Contribution	**4.7**	**3.0**	**3.6**	**3.8**	**3.5**	**3.7**
You do not own it (you cannot see or edit it)	-3.9	-4.2	-3.8	-3.2	-3.5	-3.4
You own it (you can see and edit it)	1.6	0.0	2.0	-0.5	0.2	0.1
A public community owns it (anyone can see and edit it)	0.4	0.2	0.2	0.3	0.2	0.2
Your name / username is attached to it (not anonymous)	-1.8	0.5	-1.3	-0.7	-0.6	-0.8
You receive no credit for it (it is anonymous)	0.9	0.9	0.8	1.2	1.0	1.1
Your contribution could be rejected	-0.1	-0.2	-0.2	-0.1	-0.2	-0.1
Your contribution is automatically accepted	2.9	2.8	2.3	3.1	2.9	2.9

Figure 8.7: Attribute importance per attribute is the bold number in the same row as the attribute name. This is the difference between the level with the highest and lowest relative utility level per attribute.

when comparing all everday visitors against those who only submitted accurate edits, where there is a 25–31% increase in preference share across estimated task completion times. For tasks that pay $8 per hour among those who only submitted accurate edits, the preference share for tasks that take 15 minutes or less to complete for unpaid contributors is 2 to 5 times greater than paid crowdworkers. While money can motivate accurate contributions, paid crowdworkers require larger pay levels per hour, while paying Prolific's minimum of $8 per hour to unpaid contributors should yield highly accurate edits. In summary, if pay per contribution could be introduced into a system like Drafty, that could help increase the number of accurate contributions. Next, we will analyze how much money or time can be changed to maintain the same level or preference (i.e., motivation) across different types of tasks.

Another way to measure trade-offs is to use partworth functions. We can measure the trade-offs paid crowdworkers and unpaid contributors make between two attributes by using their levels' relative utility scores to understand how to change pay levels or task times [182]. In this section, we will walk through an initial example of using pay level



Time & Pay	$0	$4	$8	$12	$16	Unpaid Contributors
1	14%	30%	29%	45%	**50%**	
5	11%	20%	18%	29%	32%	
15	12%	22%	22%	34%	36%	
30	11%	13%	9%	12%	16%	
60	6%	10%	10%	8%	10%	

Time & Pay	$0	$4	$8	$12	$16	Unpaid Contributors 100% Accurate
1	20%	23%	32%	**50%**	44%	
5	16%	23%	32%	41%	45%	
15	16%	19%	36%	43%	43%	
30	15%	7%	9%	16%	22%	
60	0%	0%	0%	5%	7%	

Time & Pay	$0	$4	$8	$12	$16	Paid Crowdworkers
1	1%	7%	14%	34%	**50%**	
5	4%	4%	8%	26%	46%	
15	4%	4%	7%	23%	42%	
30	2%	3%	4%	10%	20%	
60	3%	3%	4%	6%	10%	

Time & Pay	$0	$4	$8	$12	$16	Paid Crowdworkers 100% Accurate
1	1%	7%	13%	42%	**50%**	
5	0%	1%	16%	20%	44%	
15	0%	4%	7%	22%	33%	
30	0%	1%	0%	19%	7%	
60	0%	0%	0%	1%	0%	

Figure 8.8: This Figure shows the Preference Share when comparing the optimal crowd contribution task against different combinations of pay level and estimated time to complete across unpaid contributors and paid crowdworkers (overall and those who only submitted accurate edits). The teal cell with 50% is the optimal task per group. It is 50% because comparing Preference Share across two identical tasks would result in a perfect split of preference. People with 100% accuracy prefer shorter task times but are more willing to do 5 to 15 minute tasks for a lower pay level.

per hour and estimated time to complete a task. These two highly motivating attributes are easy to manipulate when designing new crowd contribution tasks.

How does reducing estimated task completion time affect the pay level? We have a task that pays $12 per hour and takes an estimated 30 minutes to complete. If we can reduce the estimated time to complete from 30 to 15 minutes, what is the maximum we can decrease the pay level per hour to maintain the same level of motivation for this new task that takes less time to complete?

The relative utility scores can be used to compute this trade-off. Figure 8.4 shows the relative utility scores for paid crowdworkers and unpaid contributors.

For unpaid contributors, using the relative utility scores for the estimated time to complete, we note that reducing the estimated time will increase the utility (user motivation) to complete the task by 5.5. How much should we modify the pay level per task to maintain the same level of utility? Since the pay level's relative utility scores decrease as the pay level decreases, we can look at the difference in relative utility between $12 per hour and $8 per hour (the next lowest pay level per hour). Going from $12 to $8 per hour will decrease utility by 3.8.

Equation 8.1 shows the trade-off for pay level per hour when reducing task completion time from 30 to 15 minutes for unpaid contributors to Drafty. The numerator is the difference in the relative utility scores for the two levels we will modify (i.e., estimated time to complete, 30 minutes to 15 minutes). The denominator is the difference in the relative utility scores for the two levels we will modify (i.e., estimated time to complete, 30 minutes to 15 minutes). The result of this fraction will be multiplied by the difference between the level for the attribute we want to know the tradeoff for (i.e., pay level of \$4 per hour).

$$\text{tradeoff} = \frac{|-2.0\ -\ 2.5|}{|5.0\ -\ 1.2|} \cdot \$4\ =\ \$4.74 \tag{8.1}$$

Suppose we can reduce the estimated time to complete by 15 minutes for a group of Drafty's unpaid contributors. In that case, we can also reduce the pay level per hour by a maximum of \$4.74 per hour so the new task maintains the same level of utility or motivation to the user. For paid crowdworkers following the same method, reducing estimated task completion from 30 to 15 minutes would allow us to only reduce by a maximum of \$3.35 per hour. That is because the pay level per hour has a greater effect on paid crowdworkers' motivation compared to unpaid contributors; see the relative utility scores for \$8 and \$12 per hour in Figure 8.4. For paid crowdworkers, a reduction of \$3.35 per hour from a base level of \$12 would still fit within Prolific's fair payment guidelines.

How does reducing estimated task completion time for short tasks affect the pay level? One area where the trade-offs for task completion time differ is following the Prolific guidelines and increasing an unfair pay level of \$4 per hour to \$8 per hour for a long 60 minute task. To maintain the same level of motivation as before, we could reduce task time for everyday users by around 14 minutes. For paid crowdworkers, we could reduce task time by 25 minutes. That is a 79% increase in task time saved for paid crowdworkers when increasing the pay level per hour to a fair wage based on Prolific's guidelines.

How does changing a task from boring to interesting affect the pay level per hour? Maybe a requester has had difficulty attracting paid crowdworkers for a task that appears boring even though it pays \$16 per hour. If they spent time to make the task interesting, they could decrease the pay level by a maximum of \$3.23 per hour for paid crowdworkers. Chapters 3, 5, and 7 show that unpaid contributors will make contributions matching their interests. If a task paying \$16 per hour was redesigned to be more interesting to unpaid contributors, the pay level could be reduced by a maximum of \$16 per hour. For unpaid contributors who contribute their time and knowledge, this is the effect of task perception and aligning contributions with their interests.

How does changing a task from boring to interesting affect task time? There could be a scenario where someone is designing a public data system with lengthy tasks that the average contributor finds boring. If they made a previous 15 minute task interesting instead of boring, how much longer could they expect someone to spend on the task while maintaining the same level of motivation? For unpaid contributors, the task could be 20 minutes longer, while for a paid crowdworker, the task could be 10 minutes longer. Unpaid contributors are willing to spend twice as long on an interesting task because they are more intrinsically motivated. Regardless, a requester could save time and money by increasing a task's interest level for paid crowdworkers.

How does changing a task where an unpaid contributor works alone compared to collaborating with others affect task time? Unpaid contributors have a strong preference for completing tasks where they collaborate with other people to complete them. Whereas paid crowdworkers indicated little difference between working on tasks alone or together, see Figure 8.7. Among unpaid contributors, those who only made accurate edits had a strong preference for collaboration than unpaid contributors who did not. Unpaid contributors who only made accurate edits would be willing to spend up to 18 additional minutes on a 15 minute task where they collaborate with others compared to working alone. In contrast, unpaid contributors who did not make accurate edits would be willing to spend up to 8 additional minutes on a 15 minute task where they collaborate with others compared to working alone. Unpaid contributors who only made accurate edits are willing to spend over twice as many additional minutes if the tasks involve collaboration. Thus, developing collaborative features in systems like Drafty should help groups of unpaid contributors work together to accomplish longer tasks. This might explain how groups collaborated in classic group creation activities in 6-3-5 Brainwriting [189] and C-Sketch [204].

How does collaborating with other people versus AI affect the pay level per hour for paid crowdworkers who complete tasks? In a system like Profilic, if paid crowdworkers have selected all available slots, no one else can accept that task. Therefore, you want to create crowd contribution tasks that motivate paid crowdworkers to complete their selected tasks. In our results, paid crowdworkers who edited data on Drafty preferred the possibility of collaborating with AI compared to other people. This might be because AI could ideally decrease task time and effort. Based on our previous results, introducing AI into the task to contribute could help reduce the pay level per hour. For example, if you modify a $12 per hour task that a paid crowdworker would collaborate with others to allow them to collaborate with an AI, you could decrease the pay level by a maximum of $0.90 per hour for paid crowdworkers who are more likely to edit data. Compared to paid crowdworkers who are less likely to complete the task, you could only save around $0.05 per hour. The reduction in pay level per hour is around 17 times greater for paid crowdworkers who are likely to complete the task compared to those who are not. This result shows that a system focused on recruiting paid crowdworkers could elicit more contributions if it focused on integrating AI versus other people to help the paid crowdworkers complete their tasks. This contradicts the goals of many previous paid crowdsourcing systems that focus on building a collaborative network of paid crowdworkers [64, 84]. We do not intend for others to use this result to reduce crowdworker pay but to realize how new features to create an AI-driven collaborative network can increase paid crowdworkers' motivations for completing tasks.

If an unpaid contributor can see and edit their contribution, how does that affect task time? Drafty's unpaid contributors can immediately see and edit their contributions. This provides a sense of ownership often not provided by pure paid crowdsourcing systems. While the paid crowdworkers who completed our survey preferred tasks where they could see and edit their contribution, there was no difference in the trade-offs among paid crowdworkers who only submitted accurate edits compared to those who did not, see Figure 8.7. Compared to unpaid contributors who submitted inaccurate edits, those who only made accurate edits had a stronger preference for tasks where they could see and edit their contributions. Unpaid contributors who only made accurate edits would be willing to spend up to 8 more minutes on a 15 minute task than unpaid contributors who did not make accurate edits if they could see and edit their contribution. The ability to see and edit your contribution is a core design consideration for Drafty and Sketchy.

8.3.6 Paid Crowdworkers Motivations Across Different Pay-Levels

This section compares the trade-offs paid crowdworkers make based by splitting them into three groups based on the pay level per hour they received on Prolific ($8, $12, or $16 per hour) to complete editing tasks on Drafty.

Across each group of paid crowdworkers who completed the survey, the attributes with the highest feature importance are pay level per hour, the estimated time to complete a task, and the perception per task. Overall, paid crowdworkers' motivations are similar when paying them fairly based on Prolific's guidelines of a pay level of $8, $12, or $16 per hour. Across all paid crowdworkers, there was a negative effect on their preference for crowd contributions tasks when the pay level per hour was below Prolific's minimum of $8 per hour. This result shows Prolific's pay level per hour recommendations mirror the preferences among the paid crowdworkers who completed our survey.

The relative utility scores for hypothetical tasks paying $8 per hour are relatively stable across paid crowdworkers regardless of how much we paid them to edit data on Drafty. However, these gaps increase when looking at the relative utility scores for hypothetical tasks paying $12 or $16 per hour. There is a 6.3% increase in relative utility scores comparing hypothetical tasks paying $8 between paid crowdworkers whom we paid $8 and $16 per hour. When comparing the same paid crowdworkers (paid $8 and $16 per hour), there is a 13.8% increase in relative utility scores for tasks hypothetical tasks paying $12 per hour and a 13.9% increase for tasks hypothetical tasks paying $16 per hour. The more we paid crowdworkers, the more they preferred higher paid tasks.

Results show when recruiting paid crowdworkers, the higher the pay level per hour ($8, $12, and $16 per hour) they were paid, they also preferred a slightly higher pay level. When increasing the pay level per hour they also preferred shorter tasks. There is a 6.2% increase in the attribute importance for estimated time to complete a task between paid crowdworkers paid $8 per hour versus those paid $16 per hour. If you reduced the task time from 30 to 15 minutes for a task that pays $12 per hour, what is the maximum reduction in pay level per hour per paid crowdworker that would not negatively affect their preferences? The pay level per hour could be reduced by a maximum of $3.04 per hour for crowdworkers paid $16 per hour, $3.39 per hour for crowdworkers paid $12 per hour, and $3.68 per hour for crowdworkers paid $8 per hour. The higher-paid crowdworkers are likely trying to maximize their earnings per hour [101, 108]. Table 8.3 shows the lower the pay level per hour, the more likely paid crowdworkers used the money earned from completing paid crowdsourcing tasks on Prolific as extra spending money. Overall, our results mirror prior research showing the pay level per hour is one of the most important attributes for motivating paid crowdworkers to select a task [199].

Income earned from paid crowdsourcing is used for...	$16 per hour	$12 per hour	$8 per hour
Extra spending money	50.0%	74.1%	82.1%
Helps pay some bills and expenses	40.0%	22.2%	17.9%
Helps pay the majority of bills and expenses	3.3%	0.0%	0.0%
Only source of income	6.7%	3.7%	0.0%

Table 8.3: Among the paid crowdworkers who completed the survey, the higher they were paid per hour, the more likely they relied on the money from completing tasks to pay at least part of their expenses. Notably, half of the highest paid crowdworkers ($16 per hour) relied on money earned from paid crowdsourcing to pay at least part of their expenses.

The more we paid crowdworkers per hour, the greater their level of interest per task affected their motivation. However, the less we paid crowdworkers the more they preferred tasks that helped others. Paid crowdworkers, compensated at $8 per hour, were the only group where a "boring" task did not negatively affect motivation. However, the more we paid a crowdworker, the greater the difference in the relative utility scores between boring and interesting tasks ($8 2.0, $12 4.1, and $16 4.8). For crowdworkers paid $16 per hour, the interest per task had a 2.4 times greater effect on preference and motivation than it did for crowdworkers paid $8 per hour. If we look at the trade-offs for task interest and pay level, if a task paying $12 per hour was modified to change it from boring to interesting, the pay level per hour could be reduced by a maximum of $4.08 per hour for paid crowdworkers paid $16 per hour, and by a maximum of $1.9 per hour for paid crowdworkers paid $8 per hour to maintain the same level of preference for the task. One noticeable shift for crowdworkers we paid $8 per hour compared to $16 is their preference for tasks where their contribution might help people they do not know. There is a 33% increase in their preference for tasks for this altruistic motivation. This mirrors prior work showing that some paid crowdworkers are motivated by altruistic tasks and can help others [159]. However, maybe the crowdworkers who self-select lower-paying tasks are the ones who gravitate towards tasks they feel are altruistic.

8.3.7 Unpaid Contributors are More Accurate than Paid Crowdworkers

During the study period in 2023, unpaid contributors and paid crowdworkers also added and modified data within Drafty. Table 8.4 shows that everyday unpaid contributors were more accurate across contributions to every type of data within Drafty. Notably, the accuracy of edits for a professor's join year made by unpaid contributors compared to paid crowdworkers was 1.9 times greater. Also, the accuracy of edits for a professor's subfield area of expertise made by unpaid contributors compared to paid crowdworkers was 1.5 times greater. These results mirror Chapter 5, providing more internal validity because all edits were submitted to Drafty during the same period. Overall all users of Drafty in 2023 were more likely to submit accurate edits compared to paid and unpaid users from Chapter 5. This indicates the system design decisions to decrease the time and effort to make contributions made a positive impact on paid and unpaid users of Drafty alike.

	All Edits	Add Row	Delete Row	Full Name	University	Join Year	Sub-field	Bachelors	PhD
Unpaid Contributors									
Accuracy	96%	100%	100%	98%	98%	94%	96%	87%	98%
Edits Checked	341	46	7	41	48	50	52	54	43
Paid Crowdworkers									
Accuracy	76%	73%		94%	98%	50%	63%	78%	79%
Edits Checked	340	63		46	46	46	46	46	47
Chi-Squared Test Comparing Unpaid Contributors and Paid Crowdworkers									
p-value	<0.001	<0.001		1.0	<0.001	<0.001	<0.001	0.370	<0.001
χ^2	51.5	11.3		0	0	21.3	15.1	0.8	5.9

Table 8.4: Unpaid contributors (i.e., everyday visitors) to Drafty made accurate edits at a higher rate than paid crowdworkers. Paid crowdworkers often misinterpreted a professor's subfield area of expertise and submitted nothing or the incorrect year a professor joined a university. The most common error among all users was leaving the Bachelor's blank. Reviewing individual professor's web pages it was difficult to find this information. Finding what university granted someone's Bachelor's degree often required viewing their CV or LinkedIn profile page. There were no tasks that required paid crowdworkers to delete rows of data.

Among unpaid contributors and paid crowdworkers, one common error was submitting nothing for Bachelor's degree when adding new rows of data. Hand-checking edits for Bachelor's or join year often required visiting and reviewing multiple sources, such as a professor's webpage, CV, or LinkedIn profile. Thus, the increased effort to find this information likely demotivated people to take the extra time to verify the correctness of their proposed edits. Common errors among paid crowdworkers included:

1. Submitting another year from a professor's webpage or materials as their join year.
2. Submitting blank values for where a professor received their Bachelor's degree.
3. Confusing Artificial Intelligence and Machine Learning.
4. Unable to identify a primary research area among multiple reported research areas from a professor's webpage or materials.
5. Adding non-tenure track or emeritus faculty as new rows.

Some of these errors indicate a lack of effort to keep looking for the correct information. While other errors, especially for subfields (i.e., primary research area), demonstrate a lack of domain-specific knowledge among paid crowdworkers. While paying more money might provide the incentive to keep looking for difficult to find information, directly integrating data sources into Drafty using services like WikiData or ChatGPT might prove to be more beneficial in reducing the time and effort required to verify and submit accurate edits.

Prior research shows there is a law of diminishing return when it applies to pay level and the quality of work submitted by paid crowdworkers [152]. We compensated paid crowdworkers based on the pay levels per hour ($8, $12, and $16 per hour) recommended by Prolific and enforced via their system's user interface. Among paid crowdworkers, the overall accuracy per pay level is 73% for $16 per hour, 76% for $12 per hour, and 79% for $8 per hour. When comparing the number of correct and incorrect edits, a Shapiro-Wilk test for normality reveals the data is not normally distributed. A Kruskal-Wallis H-test reveals no difference between the pay level awarded per paid crowdworker and the accuracy of that group's edits. There are also no noticeable differences across accuracy per data type among the different pay levels. These results show that while paid crowdworkers indicated increased pay levels are highly motivating, this did not result in more accurate contributions.

8.4 Discussion

The results from the discrete choice experiment align with Drafty's system design considerations and prior work on the motivation of paid crowdworkers and unpaid contributors [22, 159, 224]. There are universal attributes describing a crowd contribution that motivate everyone (helping others, voluntarily choosing to contribute, low effort, contributing specialized knowledge, etc.) are all highly motivating. However, the tradeoffs within each attribute (payment level, level of interest, etc.) varies between paid and unpaid users. For example, increasing someone's level of interest will increase motivation more if they are an unpaid contributor compared to a paid crowdworker. Even among paid crowdworkers, it can be common for the more experienced workers to have different motivations than novice workers [89]. However, our studies contain experienced crowdworkers based on Prolific's built-in pre-screening.

8.4.1 Drafty and Other Public Systems

Drafty is a public-facing system that focuses explicitly on maintaining evolving tabular data. There are many other systems where unpaid contributors complete tasks to provide and improve data quality, such as MovieLens, LabInTheWild, Reddit, Stackoverflow, Wikipedia, and WikiData.

Two public research systems that mirror the ideas of public data systems, such as freely participating contributing data, are LabInTheWild [186] and MovieLens [183]. MovieLens is another public system that mirrors many of the ideas of public data systems. Visitors can freely contribute tags that describe a movie, and they can receive movie recommendations from MovieLens. MovieLens motivates contribution through these recommendations and by showing its visitors the "value" of their contributions towards its greater community of movie lovers [183]. LabInTheWild motivates participation by allowing visitors who complete short online tests without monetary compensation to compare their results with others [186]. Beyond comparisons, LabInTheWild also motivates participation by showing how an individual's participation and contributions can help others. This public system also allows anyone to participate freely.

An area where Drafty differs from LabInTheWild and MovieLens is it is entirely anonymous. Thus, Drafty cannot directly motivate visitors to contribute by using social comparisons [44]. However, Drafty aligns with this idea of showing the value of contributions by integrating contributed data into other systems such as CS Open Rankings and Did You Know.

Moving beyond public research systems, StackOverflow and Reddit are two popular online platforms where individuals seek information and are motivated to contribute their time and knowledge. Similar to Wikipedia, these contributions often come in the form of sentences and texts. Users of StackOverflow are often motivated to contribute by providing their domain expertise or knowledge to help others [48]. Expert contributors to StackOverflow are often intrinsically motivated [144], aligning with our results showing that Drafty's highly accurate paid and unpaid contributors are also more intrinsically motivated than their less accurate counterparts. This same research by Lu et al. also highlights that gamification mechanisms do not motivate expert contributors, mirroring our results from the discrete choice experiment [144]. Compared to StackOverflow, people on Reddit answer questions that range from opinions to answers requiring domain-specific knowledge. Bogers et al. show that people who contribute to Reddit are motivated by the recreational value and customization of content, and unlike StackOverflow, they are not motivated by gamified awards [23]. Their methods to elicit preferences (i.e., motivations) use Likert scale questions. In contrast, our study uses a discrete choice experiment, which prior research shows is more effective in determining trade-offs between preferences. Unlike prior research on Reddit [23], our results show that unpaid contributors are highly motivated by collaborating with others and helping friends and family. By constructing contribution tasks that align with these ideas, unpaid contributors are likelier to spend more time on contribution tasks. While Drafty's users come from a specific demographic interested in Computer Science, other research on Reddit shows collaboration and connecting with others is a positive motivational factor [158].

Vandalism of data is a threat to the data quality for many public systems where people freely make contributions, ranging from Reddit [158] to Wikipedia and WikiData [92]. Our results for paid crowdworkers show multiple cases of people intentionally submitting bad data. For example, the professor was not a real person, or they were missing more than half of the data points. Our results for unpaid contributors showed no evidence of vandalism or laziness. The most common mistake for contributions in 2023 was not including the university where a professor received their Bachelor's degree. It is possible Drafty's everyday unpaid contributors do not vandalize the dataset because of the value of its raw data towards their community.

So while Wikipedia has been perpetually attracting unpaid contributions for longer than Drafty, people who vandalize its articles often do so because of boredom, amusement, and ideology [206]. Because Drafty is anonymous and lacks conversational features around debating edits, this eliminates the idea that vandals can engage in trolling behaviors to ease their boredom through their amusement [206, 215]. Compared to the isolated articles of Wikipedia, Drafty hosts a singular large tabular dataset of Computer Science professors. Drafty has a single focus on serving the Computer Science community; it might be that this community does not want to engage in editing behaviors that would hurt its competition (i.e., other universities). Thus explaining why Drafty's unpaid contributors do not engage in vandalism related to ideology. By building CS Open Rankings and Did You Know from this singular large dataset, they add value by sharing derived knowledge from a large tabular dataset. Thus, making it more appealing and potentially increasing unpaid contributions [154].

8.4.2 Recommendations for Developing Hybrid Paid and Unpaid Public Data Systems

When building a public data system, it is essential to find users (whether paid or not) who believe their contributions are ethical, will help others, and are interesting. They would prefer if other people could freely see and edit their contributions, and the task requires them to learn something new and allows them to contribute domain-specific knowledge. This builds on prior research showing how important intrinsic motivators and altruistic causes are to elicit high-quality contributions [160]. People are also motivated by who is asking, mirroring those by Brady et al. [28], showing that people are more likely to make accurate contributions when a friend asks. Beyond developing tasks that increase these beneficial motivators, it does subtract from the long-held idea that shorter tasks to contribute will motivate increased participation [125]. Compared to prior work, this chapter studies these attributes simultaneously using a discrete choice experiment [51, 192]. Thus, this chapter's results can compare the trade-offs made across a greater variety of motivators and ground these results by observing the real-world editing behaviors of Drafty users.

Prior research shows how mixing simple extrinsic and intrinsic rewards can improve the outcomes of tasks [31, 73]. This chapter provides evidence of attributes that universally motivate extrinsically motivated users (paid crowdworkers) and intrinsically motivated users (unpaid contributors). Building on these findings, we developed the following recommendations to help future system designers and requesters develop fair and equitable crowd contribution tasks and hybrid systems for paid crowdworkers and unpaid contributors:

1. Engage with your potential contributors to continuously understand their motivations and needs.

2. To build a community of contributors, find people with hobbies, interests, and specialized knowledge that align with your system's tasks to contribute.

3. Demonstrate how someone's contribution will help others. Immediately after someone contributes, show them an insight or the positive impact their contribution will have.

4. Communicate clearly how someone's contribution is ethical and will be used to help others and create ethical technologies and outcomes.

5. If possible, allow people to learn something new through engaging with and contributing to your system.

6. Reduce the effort to contribute to encourage unpaid contributors to freely contribute and maximize paid crowdworkers' time.

7. When asking someone to contribute, ensure a person or a public system is asking (not a bot or a company); if possible, the person asking should be known to the community.

8. If you can provide monetary compensation for a contribution, ensure the hourly pay level is a minimum of $8 per hour.

9. Paid crowdworkers will produce quality contributions when they require less effort and do not require domain-specific knowledge.

10. If you cannot provide monetary compensation for contributions, the intrinsic rewards must outweigh the lack of pay.

11. For unpaid contributors, ensure the system enhances the sense of collaboration among the community.

12. For paid crowdworkers, they might be open to AI-assisted tasks to contribute. If so, ensure the AI decreases task completion times and reduces effort.

13. Avoid adding badges, credits, or reputation points for contributions. Instead, focus on developing intrinsic rewards for contributing.

14. Keep contributions anonymous and request as little personal information as possible from contributors. If possible, request none.

15. When someone contributes, automatically accept the contribution and ensure the contributor and others can see and edit each contribution.

As a public data system, Drafty mirrors these recommendations. It has always appealed to a community of users interested in Computer Science profiles and data. Students go to Drafty to find advisors; others use its data to analyze hiring trends in different research areas. While paying crowdworkers garnered good contributions for easy-to-find and challenging to interpret data, developing low-effort methods to contribute motivated Drafty's everyday unpaid contributors to edit and maintain this type of data. Drafty accomplishes this by allowing anyone to freely come in and edit anything, creating a sense of community and collaboration. We developed CS Open Rankings and Databaits to ensure people's contributions were used towards something that benefited and engaged the larger Computer Science community. We believe these extensions are ethical and help others find advisors or evaluate schools to attend. Above all, we have modified the system over time to allow user autonomy over their actions and have only directly asked for contributions through personal means and not through automated methods or AI. In this sense, this is how we slowly built a public data system over multiple years, hopefully creating a like-minded community who care about maintaining the data because it helps them and helps others too.

8.5 Limitations

There are many possible ways to describe a crowd contribution. This research combines ideas from prior research, previous chapters of this dissertation, and a pilot study to select attributes and their associated levels to run the survey. While the paid crowdworkers we recruited could be considered expert workers because they had high acceptance rates and completed hundreds of tasks, future work could replicate our study design to answer the call for future research to study the differences in motivation between novice and expert paid crowdworkers [89].

Future work should explore additional attributes and associated levels as our motivations evolve. Likewise, every person who took the survey had some interest in Computer Science professors as they either freely visited Drafty or chose to complete a paid task posted on Prolific. This study design decision was intentional, so their stated preferences from the survey could be compared with their real-world behavior within Drafty. Future work could study how the motivations might differ based on different tasks and datasets that attract people to complete the survey.

8.6 Conclusion

This chapter shows how different attributes that describe a crowd contribution task (pay level, estimated time to complete, data ownership, helping others, interest, providing personal information, etc.) universally motivate paid crowdworkers and unpaid contributors to contribute. However, highly accurate contributors are more motivated by tasks with intrinsic qualities (helping others, their level of interest) where they freely choose to contribute, or people or public systems ask them to. Even highly accurate unpaid contributors are motivated by the prospect of receiving monetary compensation for exceptional work. These results suggest a hybrid system that mixes extrinsic and intrinsic motivators is possible and can appeal to highly accurate unpaid contributors and paid crowdworkers. This chapter also directly compares the accuracy of contributions made by paid crowdworkers and unpaid contributors using the same system, showing that unpaid contributors are more accurate while controlling for the confounding variables present in the case studies from Chapter 5.

The survey participants are everyday users of Drafty, both paid crowdworkers and unpaid contributors. This study design decision to mix and evaluate extrinsic and intrinsic motivators builds on how mixing extrinsic and intrinsic rewards can improve the outcomes of contribution tasks [31, 73]. The results show that pay level per task, estimated time to completion, and task perception (interest and potential impact of contributions) are the strongest motivators for contributing. However, the study also highlights that paid and unpaid contributors make different trade-offs when considering these motivators. Paid crowdworkers prioritize pay level and time to complete, while unpaid contributors are willing to complete longer tasks that pay less. Additionally, the study confirms that people are motivated by tasks that align with their interests and allow them to contribute their specialized knowledge to help others. Notably, participants who only made accurate edits within Drafty also preferred tasks that would pay less if they aligned with their interests and allowed them to contribute their specialized knowledge to help others. This aligns with prior paid crowdsourcing research [51] but expands this finding across everyday users of a public system. The utility scores presented in this chapter can be used by people designing crowd contribution systems for paid crowdworkers, unpaid contributors, or both. They better understand the potential positive or negative impact their system design decisions can have on their users. Hopefully, this moves us towards a fairer and more equitable future when designing crowd contribution systems.

The results of this discrete choice experiment are by no means a stopping point. This study could be conducted with users of other systems or recruited from popular platforms such as Wikipedia or StackOverflow. Do those users share the same universal motivators as Drafty's users? Do they make the same type of trade-offs when selecting tasks? These limitations should be explored in the future as our motivations evolve. Moving beyond the present, future work should explore how to streamline introducing new attributes and levels as society changes. While unpaid contributors in our study did not prefer collaborating with AI, some paid crowdworkers had a positive preference. Notably, in our pilot study in the Fall of 2022, people had a much stronger negative impression of collaborating with AI. Thus the idea that we should all be fully integrating AI into the collaborative loop is changing rapidly. This small shift in preferences highlights that as our data evolves, so will our preferences. Hence the continuous need to modify systems to match their community's evolving needs and preferences.

Chapter 9

Conclusions, Insights, and Future Directions

9.1 Insights and Conjectures

9.1.1 Theoretical Framework: Motivating People to Make Accurate Contributions

This section reviews each chapter as it relates to the theoretical framework discussed in the Introduction. How does each chapter provide evidence of how different external factors increase/decrease user motivation to contribute and the accuracy or quality of those contributions?

Chapter 3 shows initial evidence over 7 months that everyday visitors to a public system will become unpaid contributors by voluntarily making accurate contributions even when the pay level is $0 per hour. The User Interest Profile shows that when user interest is high compared to low, people will make more contributions, and those contributions are more accurate. However, this chapter shows that when a system (i.e., Drafty) asks someone for contributions, that can increase accuracy but decreases the likelihood they will contribute. The system is also designed to automatically accept someone's contribution, which shows initial evidence of increasing the frequency of contributions. Compared to later chapters, Drafty was not well known at this point. Thus it had not made a big impact on the Computer Science community. These preliminary observations showed the potential for tasks that help someone's peers or community might have on increasing accurate contributions.

Chapter 4 shows that paid crowdworkers will accept tasks to edit a tabular dataset of Computer Science profiles even when their interest is low. When data requires domain-specific knowledge, the number of contributions and accuracy decreases. When data is difficult to find or interpret, then the number of contributions and accuracy go down. This chapter also shows initial evidence that when the pay level is increased, so does accuracy. However, there is no maximum pay level that ensures 100% accurate contributions. This chapter also shows the possibility that someone might be paid for doing exceptional work (i.e., bonuses) will also increase accuracy and contributions. A lower initial pay level can be offered if there is a possibility that someone might be paid for doing exceptional work (i.e., bonuses).

Chapter 5 shows that Drafty's tabular data evolve annually. It also shows how real-world events, like PhD application season and new faculty hires, can increase motivation while the pay level stays at $0 per hour. This chapter's case study on unpaid contributors shows that when more effort and time are required to contribute (i.e., adding new rows of data), the number of contributions decreases, but accuracy remains the same. This chapter replicates the observation from Chapter 3 that when a system (i.e., Drafty) asks people for a contribution, this increases accuracy but decreases the likelihood someone contributes. This chapter's case study on paid crowdworkers shows that domain-specific knowledge gained through completing tasks can increase the accuracy of contributions. Paid strategies where the time, effort, and domain-specific knowledge to contribute are low increase accurate contributions. At the same time, paid strategies that increase the amount of time, effort, and domain-specific knowledge increase the amount of money required to get accurate contributions.

Chapter 6 studies contributions in the form of sketches that evolves quickly over four-minute sketching. This chapter shows that when self-reported domain-specific knowledge increases, the quality of contributions also increases. It shows that quality and contributions also increase when a system provides a sense of collaboration. This sense of collaboration is also supported by how the Peek feature allows users contributions to help their peers and community, increasing the quality and number of contributions. This chapter also shows that when unpaid users are provided freedom and agency to interact with a system and contribute when they want will also increase quality and contributions. The Peek feature within Sketchy allows people to learn something new while contributing, which increases the quality and number of contributions. As the sketchy system involved, the pilot studies show that developing more interesting tasks and reducing the effort to contribute also increase the quality and the number of contributions. Sketchy was also designed to create crowd contribution tasks where only the user owns the contribution (they can see and edit it), and their contributions are anonymous.

Chapter 7 builds on the lessons of Sketchy by redeveloping Drafty to reduce efforts to contribute while providing everyday unpaid contributors the opportunity to learn something new from the data while contributing. This chapter shows that reducing the effort and time required (i.e., adding new rows of data) to contribute compared to the old version of Drafty increases the accuracy and frequency of contributions. The "Did You Know" feature and CS Open Rankings show that providing insights from the source data helps users learn something new, know their contribution is helping others, provides transparency over the ethical outcomes of their contribution, and helps increase interest around the source data. All of these factors increase the accuracy and frequency of contributions. This chapter also uses these dynamic and automatic recruitment methods to increase the accuracy and frequency of contributions compared to when the old version of systems asked people to fix data. In addition, this chapter shows that asking friends or people in your community to add data increase the likelihood of them contributing and the accuracy of these contributions. These observations are made while keeping the pay level per task at $0.00 per hour over two years of natural usage of the new

Drafty system.

Chapter 8 explores the various external factors (extrinsic and intrinsic motivators) observed from the previous chapters within a discrete choice experiment. It shows that the pay level among paid crowdworkers and unpaid contributors is highly motivating but a more influential motivator for paid crowdworkers. That higher pay does not increase accuracy. In the absence of monetary compensation, unpaid contributors' intrinsic motivators will increase the accuracy of contributions. This chapter supports the previous observations that when people volunteer, or a friend asks them to complete crowd contribution tasks, they are motivated by tasks that are automatically accepted, require domain-specific knowledge, are interesting, quick to complete, require low effort, and will help others. They are also motivated when a user or a community can edit their contributions. The final versions of Drafty and Sketchy were developed and deployed over eight years. They represent many of these positive factors for increasing user motivation. Maximizing user motivation allows users of these systems to maintain data over short and long periods. When designing a new public data system, it is essential to develop crowd contribution tasks that integrate these positive motivators.

9.1.2 Recommendations for Building Public Data Systems and Crowd Contribution Tasks

This section features common insights, conjectures, and future work from this dissertation's chapters. These are my own thoughts based on my successes and failures in building publicly available data systems over eight years.

Understand the Anonymous Community of Contributors

The community of anonymous users can evolve as the data does. Continuously engaging with and listening to a public data system's community of users is essential to creating features that will sustain and improve user motivation. A critical factor in improving Drafty and Sketchy throughout this dissertation is listening to their community of users and then translating those conversations and observations into new features. For example, Drafty's users continuously emailed us to ask for the raw CSV data to create analysis and insights for themselves, their departments, or even public use. These conversations and successes motivated us to create Computer Science Open Rankings and the Databaits API. Chapter 7 describes how these add-on systems to Drafty increased visitors, their conversion rate, and the accuracy of edits compared to our previous methods. For Sketchy, users in its pilot studies provided essential feedback on reducing efforts to sketch (i.e., contribute) and requesting that sketches be randomly selected when a user initiates the Peek feature.

Future work could explore user-driven methods where they can provide anonymous feedback and ideas to help formalize this organic process of the user-to-creator communication loop. While this could use an active GitHub issues list, something more organic built into the public data system might yield different results and better insights. A simple method to better understand your anonymous community of users is to conduct a discrete choice experiment to understand people's motivations and trade-offs when selecting tasks to contribute.

Identify the Constraints of a Public Data System

There are unavoidable system and task constraints that limit users of a public data system. For example, in Drafty, we are constrained by a limited to non-existent budget. We can not pay everyday visitors who choose to contribute. Nor could we continuously pay crowdworkers over multiple years. Thus, we focused on building systems like Computer Science Open Rankings that could increase user motivation while the pay level per hour remained at $0. In Sketchy, we had to drastically reduce the effort to contribute because each sketching task lasted only 4 minutes. Identifying these constraints and then modifying the public data system to account for them will help make it sustainable over time. The utility scores from Chapter 8 can be used to identify how specific constraints (pay level, the effort to contribute, who is asking people to contribute, Etc.) can potentially increase or decrease user motivation. These results show one way to overcome the potential de-motivating factors arising from a system's constraints. Similar to the previous recommendation, if you are building a new public data system, conducting a discrete choice experiment could be beneficial to understand a new community's motivations and trade-offs when they select tasks to contribute.

Paid Crowdworkers for Bootstrapping an Evolving Dataset and Unpaid Contributors for Maintenance

A public data system has a cold start problem. It requires a dataset to bootstrap itself and attract everyday visitors. This dissertation shows strategies for paying crowdworkers to collect and bootstrap a dataset that can be cost-effective and quick. However, paid crowdworkers need more domain-specific knowledge for difficult-to-interpret data. Also, the faster a dataset evolves, the more often you must pay more crowdworkers to verify and improve the data. This is where a public data system's unpaid contributors who visit can make contributions to maintain the evolving data over time.

AI presents a new opportunity to bootstrap datasets, specifically large language models. Future research can compare their accuracy and speed in the initial collection phase and the maintenance phase with both paid crowdworkers and unpaid contributors. This research can directly influence how public data systems bootstrap initial datasets and maintain the data long-term. Regardless of these results, large language models still need more recent data. Thus there is still a future need for paid crowdworkers and unpaid contributors.

Understand your Data's Subjectivity to Improve Accuracy

The more subjective the evolving data is, the more domain-specific knowledge is required to make accurate contributions. Unpaid contributors, or everyday visitors to a public data system, often possess the domain-specific knowledge to maintain evolving data that requires domain-specific knowledge accurately.

Future work could use paid crowdworkers behaviors to understand a dataset's subjectivity. In one way, paying crowdworkers to edit or label data could be an empirical method to understand the subjectivity of specific data points. If they continuously misinterpret data, that indicates it might require someone with more expertise. Thus, if a dataset requires domain-specific knowledge, developing a public data system to attract interested everyday visitors to become unpaid contributors to improve the data is a good use case.

Recommend Randomized Examples when Data Shares a Common Theme

If you have a public dataset with a common theme: Computer Science professors, computer hardware set-ups for gaming, universities with mask mandates, or sketches of logo designs, these will likely attract users with shared interests. When this happens, you want to recommend diverse examples or information derived from the source data. Any of these examples generally align with their interests and will motivate further interactions and contributions back to the source data.

Future work should look deeper at when user interest and recommendation preferences shift. How does user interest align with the diversity of a dataset? What if Drafty's Computer Science professor dataset featured universities outside the US and Canada? Would its everyday users still prefer randomized examples?

Develop Passive Rewards within the User Community to Motivate Engagement and Contributions A public data system should reward the user visiting, engaging, and contributing to its data with something that has life and meaning outside the system. For example, learning a skill. This is in contrast to gamified systems, where the systems awards reputation points or badges for participation [153], the reward for users must be intrinsic, provide them with immediate utility, and should benefit the community of users who engage with the system. For example, CS Open Rankings automatically updates itself after a user contributes to Drafty. Thus, the updated rankings are a reward that attracts the user in the first place and is passively updated after their contribution. This reward creates a feedback cycle that benefits the user and the information. These contributions create high-quality sources of information for the community.

In the future, researchers can engage in rigorous qualitative observations with members of a public data system, chronicling how to employ human-centric design principles to build reward mechanisms from the source data. Are paid crowdworkers a viable group to provide this kind of feedback, or does it require everyay visitors to a public system?

Devote Engineering Time to Develop Features that Reduce Time and Effort to Contribute Chapter 8 shows how the perceived effort and estimated time to complete a task strongly affect the motivation of paid crowdworkers and unpaid contributors. Unlike other motivating factors such as pay level, user interest, their prior knowledge, someone designing a public data system has direct control over the time and effort required for a user to interact with the system to complete a task and contribute. This is well-spent engineering time, as these improvements will help motivate users to contribute regardless of pay level or other extrinsic rewards such as badges or reputation points.

One future idea to help reduce the effort required to contribute is to integrate AI into public data systems. There is no helpline or lengthy documentation for how to use most public data systems, as they are built for and appeal to smaller communities of users. AI can help answer questions or, more directly, teach users the domain-specific knowledge required to contribute. This dissertation shows that users abandon tasks where the correct answer or data is difficult to find. If a large language model is trained on data from the internet, then tasks requiring searching the internet could be automated to reduce the effort for paid crowdworkers and unpaid everyday users.

9.2 Conclusion and Future Directions

This dissertation develops two novel public data systems that demonstrate how to continuously develop new features and extensions to motivate everyday users to make continuous contributions over short and long periods. Since 2016, 144,117 visits made at least one interaction with Drafty's spreadsheet interface or contributed a sketch in Sketchy. This does not include visitors who visit computer science open rankings, which attracted 12 times more visitors than Drafty's spreadsheet interface based on our web server logs from February 19th to March 8th of 2023. The number of total visitors who view and read content to these public data systems is far higher than the reported 144,117 visits who made recordable interactions.

In Chapter 3, I developed Drafty, an interactive web-based spreadsheet where anyone can freely see and edit existing tabular data about Computer Science professors. This initial version of the system uses the geometry of a tabular dataset to convert a user's passive interactions into their User Interest Profile to predict their level of interest per row. Using this interest profile, Drafty asks people to review data matching their interests, showing that interest leads to more frequent and accurate contributions. Chapter 4 builds on studying how people verify and improve existing tabular data by conducting paid crowdsourcing studies. It develops new paid verification strategies by studying novice requesters' habits, successes, and failures. This chapter also shows initial evidence that increased payment per task does not directly influence the accuracy of contributions. However, the possibility of being paid extra for doing exceptional work does.

Chapter 5 builds on previous chapters by running two case studies. The first case study uses Drafty from Chapter 3 to study the accuracy and editing behaviors of Drafty's everyday unpaid contributors over two and half years. In the second case study, novice requesters use the paid verification strategies developed in Chapter 4 to study the accuracy and editing behaviors of paid crowdworkers editing the same type of tabular dataset used in Drafty. Overall, paid crowdworkers lack the domain-specific knowledge of everyday unpaid contributors when editing subjective or difficult-to-interpret data. While everyday unpaid contributors to Drafty made highly accurate contributions, especially when prompted by the system to fix data matching their interests, they often rejected edit requests from the system. They did not make contributions that required more effort, such as adding new rows of data. How could Drafty be improved to increase users' motivations, thus garnering more frequent contributions?

While I conducted the case studies on Chapter 5 I developed another public data system, Sketchy. Chapter 6 introduces Sketchy. Students in a User Interface User Experience course engage in voluntary sketching activities in this public data system. They simultaneously contribute sketches and view their peers' sketches evolve to maintain a collection of inspirational stimuli during four-minute tasks. Sketchy shows that users with higher self-reported domain-specific knowledge contribute better quality sketches. Notably, during the system development, it shows that allowing users agency over their interactions to sketch and peek at their peers' sketches leads to more engagement and contributions. When Sketchy selects a sketch to show a user, users prefer a diverse set of random sketches instead of the highest quality or most inspirational ones.

Building on these lessons from Sketchy, in Chapter 7, I develop a new version of Drafty that grants users total freedom over their interactions with the system. This new version also streamlines and reduces the effort required to edit single cells and add new rows of data. This chapter also develops new additions that use Drafty's data, CS Open Rankings (meta-ranking for Computer Science departments), and Databaits (an API to generate insights from sentence templates and aggregate statistics using tabular data). Visitors from these sources make more contributions at the same level of accuracy as asking people in the Computer Science academic community to edit Drafty's data. Also, this chapter shows Drafty's unpaid contributors prefer to see a random insight created from Drafty's six times more often than seeing a similar insight to what they have already seen. Overall, CS Open Rankings and the Databaits API provided automated methods to continuously provide updated insights about Drafty's source data and motivate everyday people to visit, engage, and contribute to the same source data.

What motivates people to contribute to public data systems like Drafty and Sketchy? Combining the quantitative results and qualitative observations from the previous chapters, Chapter 8 conducts a survey-based study, a discrete choice experiment, with Drafty's everyday users, both paid crowdworkers and unpaid contributors, to answer this question. This chapter shows that the strongest motivators for contributing are pay level per task, estimated time to completion, and task perception (someone's interest and the potential impact of their contributions). However, paid and unpaid contributors make different trade-offs. Paid crowdworkers will forfeit task perception if the pay level is high or the completion time is short. In contrast, unpaid contributors will complete longer tasks that pay less. This trade-off aligns with the prior chapters' observations, where people are universally motivated by tasks where they can contribute their specialized knowledge that align with their interests.

Using public data systems to engage anonymous users in maintaining evolving data is an innovative approach that develops features and systems that integrate individuals' intrinsic motivations to contribute to a larger goal. This goal aligns with the idea of maintaining a valuable source of data or information with real-world utility for a community of like-minded users. This approach to fairly engaging and harnessing the knowledge of communities can be further developed in the future and applied to other domains. By building systems that cater to a specific group, it is possible to streamline the development process to attract everyday visitors to become unpaid contributors in ways popular platforms cannot. Future work can compare the efforts of unpaid everyday users in systems like Drafty to contributors in large platforms such as WikiData. Future research can expand public data systems to integrate AI, such as large language models, within the user-contribution loop to augment human behaviors further. As data evolves, engaging the right user to maintain its recent data will be necessary, as large language models lack the accuracy and content for recent data.

Bibliography

[1] Hervé Abdi. Holm's sequential bonferroni procedure. *Encyclopedia of research design*, 1(8):1–8, 2010.

[2] Ibrahim Adeyanju. Generating weather forecast texts with case based reasoning. *arXiv preprint arXiv:1509.01023*, 2015.

[3] Bryan Alexander. The little spreadsheet that could, and did: crowdsourcing covid-19, higher education, data, and stories. `https://bryanalexander.org/research-topics/the-little-spreadsheet-that-could-and-did-crowdsourcing-covid-19-higher-education-data-and-stories/`, March 2020. (Accessed on 04/12/2021).

[4] Abdullah X Ali, Erin McAweeney, and Jacob O Wobbrock. Anachronism by design: Understanding young adults' perceptions of computer iconography. *International Journal of Human-Computer Studies*, 151:102599, 2021.

[5] Ali Alkhatib, Michael S Bernstein, and Margaret Levi. Examining crowd work and gig work through the historical lens of piecework. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 4599–4616, New York, NY, USA, 2017. ACM.

[6] I Elaine Allen and Christopher A Seaman. Likert scales and data analyses. *Quality progress*, 40(7):64, 2007.

[7] Salvatore Andolina, Hendrik Schneider, Joel Chan, Khalil Klouche, Giulio Jacucci, and Steven Dow. Crowdboard: augmenting in-person idea generation with real-time crowds. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*, pages 106–118, New York, NY, USA, 2017. ACM.

[8] Apache. Apache Kylin: Extreme olap engine for big data. http://kylin.apache.org/, 2015. [Online; accessed 2017-06-09].

[9] Manuela Aparicio, Fernando Bacao, and Tiago Oliveira. An e-learning theoretical framework. *Journal of Educational Technology & Society*, 19(1):292–307, 2016.

[10] Ioannis Arapakis, Mounia Lalmas, B Barla Cambazoglu, Mari-Carmen Marcos, and Joemon M Jose. User engagement in online n ews: Under the scope of sentiment, interest, affect, and gaze. *Journal of the Association for Information Science and Technology*, 65(10):1988–2005, 2014.

[11] Omer Awan and Farouk Dako. Reply: Use of parametric tests to analyze ordinal data. *Journal of nuclear medicine technology*, 46(3):318–318, 2018.

[12] Benjamin B Bederson and Alexander J Quinn. Web workers unite! addressing challenges of online laborers. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 97–106, New York, NY, USA, 2011. ACM.

[13] Roland Benabou and Jean Tirole. Intrinsic and extrinsic motivation. *The review of economic studies*, 70(3):489–520, 2003.

[14] Luca Benedetti, Holger Winnemöller, Massimiliano Corsini, and Roberto Scopigno. Painting with bob: assisted creativity for novices. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 419–428, New York, NY, USA, 2014. ACM.

[15] William Benjamin, Senthil Chandrasegaran, Devarajan Ramanujan, Niklas Elmqvist, SVN Vishwanathan, and Karthik Ramani. Juxtapoze: supporting serendipity and creative expression in clipart compositions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 341–350, New York, NY, USA, 2014. ACM.

[16] Yochai Benkler. Peer production and cooperation. In *Handbook on the Economics of the Internet*. Edward Elgar Publishing, Cheltenham, United Kingdom, 2016.

[17] Emery D. Berger. Csrankings, apr 2020.

[18] Michael Bernstein, Mike Bright, Ed Cutrell, Steven Dow, Elizabeth Gerber, Anupam Jain, and Anand Kulkarni. Micro-volunteering: helping the helpers in development. In *Proceedings of the 2013 conference on Computer supported cooperative work companion*, pages 85–88, New York, NY, USA, 2013. ACM.

[19] Michael S Bernstein, Mark S Ackerman, Ed H Chi, and Robert C Miller. The trouble with social computing systems research. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 389–398, New York, NY, USA, 2011. ACM.

[20] Michael S Bernstein, Joel Brandt, Robert C Miller, and David R Karger. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 33–42, New York, NY, USA, 2011. ACM.

[21] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. In *UIST '10 Proceedings of the 23rd annual ACM symposium on User interface Software and Technology*, pages 313–322, New York, NY, USA, 2010. ACM Press, ACM.

[22] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. *Communications of the ACM*, 58(8):85–94, 2015.

[23] Toine Bogers and Rasmus Nordenhoff Wernersen. How'social'are social news sites? exploring the motivations for using reddit. com. *iConference 2014: Breaking Down Walls: Culture-Context-Computing*, pages 329–344, 2014.

[24] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, New York, NY, USA, 2008. ACM.

[25] Michelle A Borkin, Azalea A Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2306–2315, 2013.

[26] Leon Bornemann, Tobias Bleifuß, Dmitri V Kalashnikov, Felix Naumann, and Divesh Srivastava. Natural key discovery in wikipedia tables. In *Proceedings of The Web Conference 2020*, pages 2789–2795, New York, NY, USA, 2020. ACM.

[27] Ria Mae Borromeo and Motomichi Toyama. An investigation of unpaid crowdsourcing. *Human-centric Computing Information Sciences*, 6(1):68:1–68:19, December 2016.

[28] Erin Brady, Meredith Ringel Morris, and Jeffrey P Bigham. Social microvolunteering: Donating access to your friends for charitable microwork. In *Second AAAI Conference on Human Computation and Crowdsourcing*, pages 8–9, Menlo Park, CA, USA, 2014. AAAI.

[29] Erin Brady, Meredith Ringel Morris, and Jeffrey P Bigham. Gauging receptiveness to social microvolunteering. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1055–1064, New York, NY, USA, 2015. ACM.

[30] Robin Brewer, Meredith Ringel Morris, and Anne Marie Piper. " why would anybody do this?" understanding older adults' motivations and challenges in crowd work. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 2246–2257, New York, NY, USA, 2016. ACM.

[31] Gordon Burtch, Yili Hong, Ravi Bapna, and Vladas Griskevicius. Stimulating online reviews by combining financial incentives and social norms. *Management Science*, 64(5):2065–2082, 2018.

[32] Bill Buxton. *Sketching user experiences: getting the design right and the right design*. Morgan Kaufmann, Burlington, MA, USA, 2010.

[33] Zoya Bylinskii, Nam Wook Kim, Peter O'Donovan, Sami Alsheikh, Spandan Madan, Hanspeter Pfister, Fredo Durand, Bryan Russell, and Aaron Hertzmann. Learning visual importance for graphic designs and data visualizations. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 57–69, New York, NY, USA, 2017. ACM.

[34] Luca Cacciapuoti, Veselin B Kostov, Marc Kuchner, Elisa V Quintana, Knicole D Colón, Jonathan Brande, Susan E Mullally, Quadry Chance, Jessie L Christiansen, John P Ahlers, et al. The tess triple-9 catalog: 999 uniformly vetted exoplanet candidates. *Monthly Notices of the Royal Astronomical Society*, 513(1):102–116, 2022.

[35] Ana Caraban, Evangelos Karapanos, Daniel Gonçalves, and Pedro Campos. 23 ways to nudge: A review of technology-mediated nudging in human-computer interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–15, New York, NY, USA, 2019. ACM.

[36] Dana R Carney and Mahzarin R Banaji. First is best. *PloS one*, 7(6):e35088, 2012.

[37] Erin A Carroll, Celine Latulipe, Richard Fung, and Michael Terry. Creativity factor evaluation: towards a standardized survey metric for creativity support. In *Proceedings of the seventh ACM conference on Creativity and cognition*, pages 127–136, New York, NY, USA, 2009. ACM.

[38] Joel Chan, Steven Dang, and Steven P Dow. Improving crowd innovation with expert facilitation. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1223–1235, New York, NY, USA, 2016. ACM.

[39] Philip K Chan. A non-invasive learning approach to building web user profiles. In *KDD-99 Workshop on Web Usage Analysis and User Profiling*, page 6, Pennsylvania State University University Park, PA, USA, 1999. Citeseer.

[40] Yolande E. Chan and Veda C. Storey. The use of spreadsheets in organizations: Determinants and consequences. *Information & Management*, 31(3):119–134, 1996.

[41] Shuo Chang, F Harper, Lingfei He, and Loren Terveen. Crowdlens: experimenting with crowd-powered recommendation and explanation. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10, pages 52–61, 2016.

[42] Sarah E Chasins, Maria Mueller, and Rastislav Bodik. Rousillon: Scraping distributed hierarchical web data. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 963–975, New York, NY, USA, 2018. ACM.

[43] Tatiana Chemi, Julie Borup Jensen, and Lone Hersted. *Behind the scenes of artistic creativity: Processes of learning, creating and organising*. Peter Lang, Bern, Switzerland, 2015.

[44] Yan Chen, F Maxwell Harper, Joseph Konstan, and Sherry Xin Li. Social comparisons and contributions to online communities: A field experiment on movielens. *American Economic Review*, 100(4):1358–1398, 2010.

[45] Ye Chen, Dmitry Pavlov, and John F Canny. Large-scale behavioral targeting. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 209–218, New York, NY, USA, 2009. ACM.

[46] Justin Cheng, Jaime Teevan, and Michael S Bernstein. Measuring crowdsourcing effort with error-time curves. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1365–1374, New York, NY, USA, 2015. ACM.

[47] Erin Cherry and Celine Latulipe. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(4):21, 2014.

[48] Morakot Choetkiertikul, Daniel Avery, Hoa Khanh Dam, Truyen Tran, and Aditya Ghose. Who will answer my question on stack overflow? In *2015 24th Australasian software engineering conference*, pages 155–164. IEEE, 2015.

[49] John Joon Young Chung, Jean Y Song, Sindhu Kutty, Sungsoo Hong, Juho Kim, and Walter S Lasecki. Efficient elicitation approaches to estimate collective crowd answers. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–25, 2019.

[50] Dennis L Clason and Thomas J Dormody. Analyzing data measured by individual likert-type items. *Journal of Agricultural Education*, 35:4, 1994.

[51] Aaron Clauset, Samuel Arbesman, and Daniel B Larremore. Systematic inequality and hierarchy in faculty hiring networks. *Science Advances*, 1(1):e1400005, 2015.

[52] Joanna Coast and Sue Horrocks. Developing attributes and levels for discrete choice experiments using qualitative methods. *Journal of health services research & policy*, 12(1):25–30, 2007.

[53] Dan Cosley, Dan Frankowski, Loren Terveen, and John Riedl. Suggestbot: Using intelligent task routing to help people find work in wikipedia. In *Proceedings of the 12th International Conference on Intelligent User Interfaces*, IUI '07, pages 32–41, New York, NY, USA, 2007. ACM.

[54] Arthur J Cropley. Fostering creativity in the classroom: General principles. *The creativity research handbook*, 1(84.114):1–46, 1997.

[55] Nigel Cross. Expertise in design: an overview. *Design studies*, 25(5):42–441, 2004.

[56] Crunchbase Inc. Crunchbase accelerates innovation by bringing together data on companies and the people behind them. https://www.crunchbase.com/, 2007. [Online; accessed 2016-08-20].

[57] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys (CSUR)*, 51(1):1–40, 2018.

[58] Deanna P. Dannels and Kelly Norris Martin. Critiquing critiques: A genre analysis of feedback across novice to expert design studios. *J. Business and Technical Communication*, 22(2):135–159, 2008.

[59] Paramita Das, Bhanu Prakash Reddy Guda, Debajit Chakraborty, Soumya Sarkar, and Animesh Mukherjee. When expertise gone missing: Uncovering the loss of prolific contributors in wikipedia. *arXiv preprint arXiv:2109.09979*, 2021.

[60] Nediyana Daskalova, Eindra Kyi, Kevin Ouyang, Arthur Borem, Sally Chen, Sung Hyun Park, Nicole Nugent, and Jeff Huang. Self-e: Smartphone-supported guidance for customizable self-experimentation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–13, New York, NY, USA, 2021. ACM.

[61] Victor De Boer, Michiel Hildebrand, Lora Aroyo, Pieter De Leenheer, Chris Dijkshoorn, Binyam Tesfa, and Guus Schreiber. Nichesourcing: harnessing the power of crowds of experts. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 16–20, New York, NY, USA, 2012. Springer.

[62] Victor De Boer, Michiel Hildebrand, Lora Aroyo, Pieter De Leenheer, Chris Dijkshoorn, Binyam Tesfa, and Guus Schreiber. Nichesourcing: harnessing the power of crowds of experts. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 16–20, New York, NY, USA, 2012. Springer.

[63] Edward L Deci and Richard M Ryan. A motivational approach to self: Integration in personality. *In Nebraska Symposium on Motivation: Perspectives on Motivation*, 38:237–288, 1991.

[64] Greg d'Eon, Joslin Goh, Kate Larson, and Edith Law. Paying crowd workers for collaborative work. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–24, 2019.

[65] Anhai Doan, Raghu Ramakrishnan, and Alon Y Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.

[66] Steven P. Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L. Schwartz, and Scott R. Klemmer. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM TOCHI*, 17(4):18:1–18:24, 2010.

[67] Anca Dumitrache. Crowdsourcing disagreement for collecting semantic annotation. In *Proc. ESWC*, pages 701–710, New York, NY, USA, 2015. Springer.

[68] Serge Egelman, Ed H. Chi, and Steven Dow. *Ways of Knowing in HCI*, chapter Crowdsourcing in HCI Research, pages 267–289. Springer New York, New York, NY, 2014.

[69] Shaoyang Fan, Ujwal Gadiraju, Alessandro Checco, and Gianluca Demartini. Crowdco-op: Sharing risks and rewards in crowdsourcing. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–24, 2020.

[70] Siamak Faridani, Björn Hartmann, and Panagiotis G. Ipeirotis. What's the right price? pricing tasks for finishing on time. In *Proceedings of the 11th AAAI Conference on Human Computation*, AAAIWS'11-11, pages 26–31, Menlo Park, CA, USA, 2011. AAAI.

[71] Jennifer Fernquist, Tovi Grossman, and George Fitzmaurice. Sketch-sketch revolution: An engaging tutorial system for guided sketching and application learning. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 373–382, New York, NY, USA, 2011. ACM.

[72] Besnik Fetahu, Avishek Anand, and Maria Koutraki. Tablenet: An approach for determining fine-grained relations for wikipedia tables. In *The World Wide Web Conference*, pages 2736–2742, New York, NY, USA, 2019. ACM.

[73] Claudia Flores-Saviaga, Ricardo Granados, Liliana Savage, Lizbeth Escobedo, and Saiph Savage. Understanding the complementary nature of paid and volunteer crowds for content creation. *Avances en Interacción Humano-Computadora*, 1(1):37–44, 2020.

[74] Mary Jo Foley. About that 1 billion microsoft office figure... https://www.zdnet.com/article/about-that-1-billion-microsoft-office-figure/, 2010. [Online; accessed 2020-04-20].

[75] Maksym Gabielkov, Arthi Ramachandran, Augustin Chaintreau, and Arnaud Legout. Social clicks: What and who gets read on twitter? In *Proceedings of the 2016 ACM SIGMETRICS international conference on measurement and modeling of computer science*, pages 179–192, 2016.

[76] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 1631–1640, New York, NY, USA, 2015. Association for Computing Machinery.

[77] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2:100–126, 2021.

[78] R Stuart Geiger and Aaron Halfaker. Using edit sessions to measure participation in wikipedia. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 861–870, 2013.

[79] Yotam Gingold, Etienne Vouga, Eitan Grinspun, and Haym Hirsh. Diamonds from the rough: Improving drawing, painting, and singing via crowdsourcing. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, Palo Alto, CA, USA, 2012. AAAI.

[80] Ilka H Gleibs. Are all "research fields" equal? rethinking practice for the use of data from crowdsourcing market places. *Behavior Research Methods*, 49(4):1333–1342, 2017.

[81] Kosa Goucher-Lambert and Jonathan Cagan. Crowdsourcing inspiration: Using crowd generated inspirational stimuli to support designer ideation. *Design Studies*, 61:1–29, 2019.

[82] Nitesh Goyal, Gilly Leshed, Dan Cosley, and Susan R Fussell. Effects of implicit sharing in collaborative analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 129–138, New York, NY, USA, 2014. ACM.

[83] Nitesh Goyal, Gilly Leshed, and Susan R Fussell. Leveraging partner's insights for distributed collaborative sensemaking. In *Proceedings of the 2013 conference on Computer supported cooperative work companion*, pages 15–18, New York, NY, USA, 2013. ACM.

[84] Mary L. Gray, Siddharth Suri, Syed Shoaib Ali, and Deepti Kulkarni. The crowd is a collaborative network. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, CSCW '16, pages 134–147, New York, NY, USA, 2016. ACM.

[85] Philipp Gutheim and Björn Hartmann. Fantasktic: Improving quality of results for novice crowdsourcing users. *EECS Dept., Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2012-112*, 2012.

[86] Daniel Haas, Jason Ansel, Lydia Gu, and Adam Marcus. Argonaut: Macrotask crowdsourcing for complex data processing. *Proceedings of the VLDB Endowment*, 8(12):1642–1653, August 2015.

[87] Daniel Haas, Sanjay Krishnan, Jiannan Wang, Michael J. Franklin, and Eugene Wu. Wisteria: Nurturing scalable data cleaning infrastructure. *Proceedings of the VLDB Endowment*, 8(12):2004–2007, August 2015.

[88] Aaron Halfaker, R Stuart Geiger, Jonathan T Morgan, and John Riedl. The rise and decline of an open collaboration system: How wikipedia's reaction to popularity is causing its decline. *American Behavioral Scientist*, 57(5):664–688, 2013.

[89] Benjamin V Hanrahan, Anita Chen, JiaHua Ma, Ning F Ma, Anna Squicciarini, and Saiph Savage. The expertise involved in deciding which hits are worth doing on amazon mechanical turk. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–23, 2021.

[90] REM Harding. An anatomy of inspiration: And an essay on the creative mood. w. heffer, 1948.

[91] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, Amsterdam, Netherlands, 1988.

[92] Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. Vandalism detection in wikidata. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 327–336, New York, NY, USA, 2016. ACM.

[93] Thomas Herndon, Michael Ash, and Robert Pollin. Does high public debt consistently stifle economic growth? a critique of reinhart and rogoff. *Cambridge journal of economics*, 38(2):257–279, 2014.

[94] Jim Hewitt and Marlene Scardamalia. Design principles for distributed knowledge building processes. *Educational psychology review*, 10(1):75–96, 1998.

[95] Daniel Hienert, Dagmar Kern, Matthew Mitsui, Chirag Shah, and Nicholas J Belkin. Reading protocol: Understanding what has been read in interactive information retrieval tasks. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, CHIIR '19, pages 73–81, New York, NY, USA, 2019. ACM.

[96] Benjamin Mako Hill and Aaron Shaw. Wikipedia and the end of open collaboration. *Wikipedia*, 20, 2020.

[97] Matthias Hirth, Tobias Hoßfeld, and Phuoc Tran-Gia. Analyzing costs and accuracy of validation mechanisms for crowdsourcing platforms. *Mathematical and Computer Modelling*, 57(11):2918–2932, 2013.

[98] Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. Incentivizing high quality crowdwork. In *Proceedings of the 24th International Conference on World Wide Web*, pages 419–429, New York, NY, USA, 2015. ACM.

[99] Orland Hoeber, Anoop Sarkar, Andrei Vacariu, Max Whitney, Manali Gaikwad, and Gursimran Kaur. Evaluating the value of lensing wikipedia during the information seeking process. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, pages 77–86, New York, NY, USA, 2017. ACM.

[100] Reid Holmes, Robert J Walker, and Gail C Murphy. Approximate structural context matching: An approach to recommend relevant examples. *IEEE Transactions on Software Engineering*, 32(12):952–970, 2006.

[101] John Joseph Horton and Lydia B Chilton. The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, pages 209–218, New York, NY, USA, 2010. ACM.

[102] Jeff Howe. The rise of crowdsourcing. http://www.wired.com/2006/06/crowds/, 2006. [Online; accessed 2020-04-20].

[103] Jeff Howe. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Crown Publishing Group, New York, NY, USA, 2008.

[104] Jeff Huang, Ryen W. White, and Susan Dumais. No clicks, no problem: Using cursor movements to understand and improve search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1225–1234, New York, NY, USA, 2011. ACM.

[105] Shih-Wen Huang and Wai-Tat Fu. Enhancing reliability using peer consistency evaluation in human computation. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 639–648, New York, NY, USA, 2013. ACM.

[106] Ting-Hao Kenneth Huang and Jeffrey P Bigham. A 10-month-long deployment study of on-demand recruiting for low-latency crowdsourcing. In *HCOMP*, pages 61–70, Menlo Park, CA, USA, 2017. AAAI.

[107] Emmanuel Iarussi, Adrien Bousseau, and Theophanis Tsandilas. The drawing assistant: Automated drawing guidance and feedback from photographs. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 183–192, New York, NY, USA, 2013. ACM.

[108] Kazushi Ikeda and Michael S Bernstein. Pay it backward: Per-task payments on crowdsourcing platforms reduce productivity. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4111–4121, 2016.

[109] Chiaki Ishiguro and Takeshi Okada. *How can inspiration be encouraged in art learning.* River Publishers, Denmark, 2018.

[110] David G. Jansson and Steven M. Smith. Design fixation. *Design Stud.*, 12(1):3–11, 1991.

[111] Hyeonsu B. Kang, Gabriel Amoako, Neil Sengupta, and Steven P. Dow. Paragon: An online gallery for enhancing design feedback with visual examples. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 606, New York, NY, USA, 2018. ACM.

[112] Keiko Katsuragawa, Qi Shu, and Edward Lank. Pledgework: Online volunteering through crowdwork. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–11, New York, NY, USA, 2019. ACM.

[113] James C Kaufman and Ronald A Beghetto. Beyond big and little: The four c model of creativity. *Review of general psychology*, 13(1):1–12, 2009.

[114] Gabriella Kazai. *In Search of Quality in Crowdsourcing for Search Engine Evaluation*, pages 165–176. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[115] Swarna Keshavabhotla, Blake Williford, Shalini Kumar, Ethan Hilton, Paul Taele, Wayne Li, Julie Linsey, and Tracy Hammond. Conquering the cube: learning to sketch primitives in perspective with an intelligent tutoring system. In *Proceedings of the Symposium on Sketch-Based Interfaces and Modeling*, page 2, New York, NY, USA, 2017. ACM.

[116] Warut Khern-am nuai, Karthik Kannan, and Hossein Ghasemkhani. Extrinsic versus intrinsic rewards for contributing reviews in an online platform. *Information Systems Research*, 29(4):871–892, 2018.

[117] Johannes Kiesel, Martin Potthast, Matthias Hagen, and Benno Stein. Spatio-temporal analysis of reverted wikipedia edits. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 122–131, Menlo Park, CA, USA, 2017. AAAI.

[118] Jinmook Kim, Douglas W Oard, and Kathleen Romanik. Using Implicit Feedback for User Modeling in Internet and Intranet Searching. *University of Maryland CLIS Technical Report*, pages 1–21, 2000.

[119] Joy Kim, Maneesh Agrawala, and Michael S Bernstein. Mosaic: designing online creative communities for sharing works-in-progress. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 246–258, New York, NY, USA, 2017. ACM.

[120] Juho Kim et al. *Learnersourcing: improving learning with collective learner activity*. PhD thesis, Massachusetts Institute of Technology, 2015.

[121] Juho Kim, Philip J Guo, Carrie J Cai, Shang-Wen Daniel Li, Krzysztof Z Gajos, and Robert C Miller. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 563–572, New York, NY, USA, 2014. ACM.

[122] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 453–456, New York, NY, USA, 2008. ACM.

[123] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, pages 1301–1318, New York, NY, USA, 2013. ACM.

[124] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E. Kraut. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 43–52, New York, NY, USA, 2011. ACM.

[125] Thomas Kohler. Crowdsourcing-based business models: how to create and capture value. *California Management Review*, 57(4):63–84, 2015.

[126] Flip Korn, Xuezhi Wang, You Wu, and Cong Yu. Automatically generating interesting facts from wikipedia tables. In *Proceedings of the 2019 International Conference on Management of Data*, pages 349–361, New York, NY, USA, 2019. ACM.

[127] Michel Krieger, Emily Margarete Stark, and Scott R. Klemmer. Coordinating tasks on the commons: Designing for personal goals, expertise and serendipity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1485–1494, New York, NY, USA, 2009. ACM.

[128] W Kuhfeld. Multinomial logit, discrete choice modeling. *An introduction to designing choice experiments, and collecting, processing and analyzing choice data with SAS. SAS Institute TS-643*, 2001.

[129] Anand Kulkarni, Matthew Can, and Björn Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1003–1012, New York, NY, USA, 2012. ACM.

[130] Chinmay Kulkarni, Steven P Dow, and Scott R Klemmer. Early and repeated exposure to examples improves creative work. In *Design thinking research*, pages 49–62. Springer, New York, NY, USA, 2014.

[131] Srijan Kumar, Francesca Spezzano, and VS Subrahmanian. Vews: A wikipedia vandal early warning system. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 607–616, New York, NY, USA, 2015. ACM.

[132] Natalie Kupferberg and Bridget McCrate Protus. Accuracy and completeness of drug information in wikipedia: an assessment. *Journal of the Medical Library Association: JMLA*, 99(4):310, 2011.

[133] Marjan Laal and Seyed Mohammad Ghodsi. Benefits of collaborative learning. *Procedia-social and behavioral sciences*, 31:486–490, 2012.

[134] Walter S. Lasecki, Juho Kim, Nicholas Rafter, Onkur Sen, Jeffrey P. Bigham, and Michael S. Bernstein. 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*, pages 1925–1934, New York, NY, USA, 2015. ACM.

[135] Sang Won Lee. Hybrid use of asynchronous and synchronous interaction for collaborative creation. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 95–98, New York, NY, USA, 2017. ACM.

[136] Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen. Shadowdraw: Real-time user guidance for freehand drawing. *ACM Transactions on Graphics (TOG)*, 30(4):27:1–27:9, 2011.

[137] Hongwei Li, Bo Zhao, and Ariel Fuxman. The wisdom of minority: discovering and targeting the right group of workers for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pages 165–176, New York, NY, USA, 2014. ACM.

[138] Sascha Lichtenberg, Tim-Benjamin Lembcke, Mattheus Brening, Alfred Benedikt Brendel, and Simon Trang. Can gamification lead to increase paid crowdworkers output? In *Wirtschaftsinformatik (Zentrale Tracks)*, pages 1188–1202, Illinois, USA, 2020. AIS.

[139] Alex Limpaecher, Nicolas Feltman, Adrien Treuille, and Michael Cohen. Real-time drawing assistance through crowdsourcing. *ACM Transactions on Graphics (TOG)*, 32(4):54:1–54:8, 2013.

[140] Tracy Xiao Liu, Jiang Yang, Lada A Adamic, and Yan Chen. Crowdsourcing with all-pay auctions: A field experiment on taskcn. *Management Science*, 60(8):2020–2037, 2014.

[141] Dastyni Loksa, Nicolas Mangano, Thomas D LaToza, and André van der Hoek. Enabling a classroom design studio with a collaborative sketch design tool. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 1073–1082, Piscataway, NJ, USA, 2013. IEEE.

[142] Jordan J Louviere and Emily Lancsar. Choice experiments in health: the good, the bad, the ugly and toward a brighter future. *Health Economics, Policy and Law*, 4(4):527–546, 2009.

[143] Kuang-Chen Lu, Ben Greenman, and Shriram Krishnamurthi. Types for tables: A language design benchmark. *Art Sci. Eng. Program.*, 6(2):8, 2022.

[144] Yao Lu, Xinjun Mao, Minghui Zhou, Yang Zhang, Zude Li, Tao Wang, Gang Yin, and Huaimin Wang. Motivation under gamification: An empirical study of developers' motivations and contributions in stack overflow. *IEEE Transactions on Software Engineering*, 48(12):4947–4963, 2021.

[145] Roman Lukyanenko, Jeffrey Parsons, Yolanda F Wiersma, and Mahed Maddah. Expecting the unexpected: Effects of data collection design choices on the quality of crowdsourced user-generated content. *MIS Quarterly*, 43(2):623–648, 2019.

[146] Yuan Luo. Evaluating the state of the art in missing data imputation for clinical data. *Briefings in Bioinformatics*, 23(1):bbab489, 2022.

[147] Eddy Maddalena, Luis-Daniel Ibáñez, Neal Reeves, and Elena Simperl. Qrowdsmith: Enhancing paid microtask crowdsourcing with gamification and furtherance incentives. *ACM Transactions on Intelligent Systems and Technology*, 2023.

[148] Joy Mahapatra, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. Statistical natural language generation from tabular non-textual data. In *Proceedings of the 9th International Natural Language Generation conference*, pages 143–152, 2016.

[149] Lindsay J Mangham, Kara Hanson, and Barbara McPake. How to do (or not to do)… designing a discrete choice experiment for application in a low-income country. *Health policy and planning*, 24(2):151–158, 2009.

[150] Andrew Mao, Ece Kamar, Yiling Chen, Eric Horvitz, Megan E Schwamb, Chris J Lintott, and Arfon M Smith. Volunteering versus work for pay: Incentives and tradeoffs in crowdsourcing. In *First AAAI conference on human computation and crowdsourcing*, pages 94–102, Menlo Park, CA, USA, 2013. AAAI.

[151] Adam Marcus. How i learned to stop worrying and love the crowd, 2013. (Online; accessed on 2020-04-20).

[152] Winter Mason and Duncan J. Watts. Financial incentives and the "performance of crowds". *ACM SigKDD Explorations Newsletter*, 11(2):77–85, 2009.

[153] Arpit Merchant, Daksh Shah, Gurpreet Singh Bhatia, Anurag Ghosh, and Ponnurangam Kumaraguru. Signals matter: understanding popularity and impact of users on stack overflow. In *The World Wide Web Conference*, pages 3086–3092, New York, NY, USA, 2019. ACM.

[154] Paweł Mikołajczak and Piotr Bajak. Does ngos' commercialization affect volunteer work? the crowding out or crowding in effect. *Public Organization Review*, 21(1):103–118, 2021.

[155] Joshua D Miller, Michael Crowe, Brandon Weiss, Jessica L Maples-Keller, and Donald R Lynam. Using online, crowdsourcing platforms for data collection in personality disorder research: The example of amazon's mechanical turk. *Personality Disorders: Theory, Research, and Treatment*, 8(1):26, 2017.

[156] Marc Miquel-Ribé, Cristian Consonni, and David Laniado. Wikipedia editor drop-off. *Wiki Workshop at The Web Conference 2021*, 2021.

[157] Laurent Moccozet, Camille Tardy, Wanda Opprecht, and Michel Léonard. Gamification-based assessment of group work. In *2013 International Conference on Interactive Collaborative Learning (ICL)*, pages 171–179. IEEE, 2013.

[158] Carrie Moore and Lisa Chuang. Redditors revealed: Motivational factors of the reddit community. *Proceedings of the 50th Hawaii International Conference on System Sciences*, pages 1–10, 2017.

[159] Meredith Ringel Morris, Jeffrey P. Bigham, Robin Brewer, Jonathan Bragg, Anand Kulkarni, Jessie Li, and Saiph Savage. Subcontracting microwork. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1867–1876, New York, NY, USA, 2017. ACM.

[160] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. What do people ask their social networks, and why?: A survey study of status message q&a behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1739–1748, New York, NY, USA, 2010. ACM.

[161] Felix Müller-Wienbergen, Oliver Müller, Stefan Seidel, and Jörg Becker. Leaving the beaten tracks in creative work–a design theory for systems that support convergent and divergent thinking. *Journal of the Association for Information Systems*, 12(11):2, 2011.

[162] Babak Naderi. *Motivation of workers on microtask crowdsourcing platforms*. Springer, New York, NY, USA, 2018.

[163] Robbie T Nakatsu, Elissa B Grossman, and Charalambos L Iacovou. A taxonomy of crowdsourcing based on task complexity. *Journal of Information Science*, 40(6):823–834, 2014.

[164] Khushnood Z Naqshbandi, Chunfeng Liu, Silas Taylor, Renee Lim, Naseem Ahmadpour, and Rafael Calvo. "i am most grateful." using gratitude to improve the sense of relatedness and motivation for online volunteerism. *International Journal of Human–Computer Interaction*, 36(14):1325–1341, 2020.

[165] Tricia J Ngoon, C Ailie Fraser, Ariel S Weingarten, Mira Dontcheva, and Scott Klemmer. Interactive guidance techniques for improving creative feedback. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 55, New York, NY, USA, 2018. ACM.

[166] Phuc Nguyen, Ikuya Yamada, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. Mtab4wikidata at semtab 2020: Tabular data annotation with wikidata. In *SemTab@ ISWC*, pages 86–95, New York, NY, USA, 2020. ACM.

[167] Geoff Norman. Likert scales, levels of measurement and the "laws" of statistics. *Advances in health sciences education*, 15(5):625–632, 2010.

[168] Victoria C Oleynick, Todd M Thrash, Michael C LeFew, Emil G Moldovan, and Paul D Kieffaber. The scientific study of inspiration in the creative process: challenges and opportunities. *Frontiers in human neuroscience*, 8:436, 2014.

[169] Jasper Oosterman, Archana Nottamkandath, Chris Dijkshoorn, Alessandro Bozzon, Geert-Jan Houben, and Lora Aroyo. Crowdsourcing knowledge-intensive tasks in cultural heritage. In *Proceedings of the 2014 ACM Conference on Web Science*, WebSci '14, pages 267–268, New York, NY, USA, 2014. ACM.

[170] Bryan Orme. Interpreting conjoint analysis data. *Sawtooth Software Research Paper Series*, 2002.

[171] Katherine Panciera, Aaron Halfaker, and Loren Terveen. Wikipedians are born, not made: a study of power editors on wikipedia. In *Proceedings of the ACM 2009 international conference on Supporting group work*, pages 51–60, 2009.

[172] Raymond R Panko. What we know about spreadsheet errors. *Journal of Organizational and End User Computing (JOEUC)*, 10(2):15–21, 1998.

[173] Alexandra Papoutsaki, Hua Guo, Danae Metaxa-Kakavouli, Connor Gramazio, Jeff Rasley, Wenting Xie, Guan Wang, and Jeff Huang. Crowdsourcing from scratch: A pragmatic experiment in data collection by novice requesters. In *Proceedings of the Third AAAI Conference on Human Computation and Crowdsourcing*, HCOMP, pages 140–149, Menlo Park, CA, USA, 2015. AAAI.

[174] Alexandra Papoutsaki, James Laskey, and Jeff Huang. Searchgazer: Webcam eye tracking for remote studies of web search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, pages 17–26, New York, NY, USA, 2017. ACM.

[175] Hyunjung Park and Jennifer Widom. Crowdfill: Collecting structured data from the crowd. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 577–588, New York, NY, USA, 2014. ACM.

[176] Cecil Piya, Senthil Chandrasegaran, Niklas Elmqvist, Karthik Ramani, et al. Co-3deator: A team-first collaborative 3d design ideation tool. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 6581–6592, New York, NY, USA, 2017. ACM.

[177] Wikimedia Project. Wikipedia and wikidata tools - meta. https://meta.wikimedia.org/wiki/Wikipedia_and_Wikidata_Tools, April 2021. (Accessed on 07/10/2021).

[178] Sihang Qiu, Ujwal Gadiraju, and Alessandro Bozzon. Improving worker engagement through conversational microtask crowdsourcing. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, New York, NY, USA, 2020. ACM.

[179] Giovanni Quattrone, Martin Dittus, and Licia Capra. Work always in progress: Analysing maintenance practices in spatial crowd-sourced datasets. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, pages 1876–1889, New York, NY, USA, 2017. ACM.

[180] Alexander J. Quinn and Benjamin B. Bederson. Human computation: A survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 1403–1412, New York, NY, USA, 2011. Association for Computing Machinery.

[181] Sam Ransbotham and Gerald C Kane. Membership turnover and collaboration success in online communities: Explaining rises and falls from grace in wikipedia. *Mis Quarterly*, 35:613–627, 2011.

[182] Vithala R Rao. *Applied conjoint analysis*. Springer Science & Business Media, 2014.

[183] Al M Rashid, Kimberly Ling, Regina D Tassone, Paul Resnick, Robert Kraut, and John Riedl. Motivating participation by displaying the value of contribution. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 955–958, 2006.

[184] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 127–134, New York, NY, USA, 2002. ACM.

[185] Michael Redmond and Alok Baveja. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, 141(3):660–678, 2002.

[186] Katharina Reinecke and Krzysztof Z Gajos. Labinthewild: Conducting large-scale online experiments with uncompensated samples. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 1364–1378, New York, NY, USA, 2015. ACM.

[187] Yuqing Ren, Sara Kiesler, and Susan R Fussell. Multiple group coordination in complex and dynamic task environments: Interruptions, coping mechanisms, and technology recommendations. *Journal of Management Information Systems*, 25(1):105–130, 2008.

[188] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, New York, NY, USA, 1994. ACM.

[189] B Rhorbach. Kreative nach regeln: Methode 635, eine neue technik zum losen von problemen. *Absatzwirtschaft*, 12:73–75, 1969.

[190] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems: Introduction and challenges. In *Recommender Systems Handbook*, pages 1–34. Springer, New York, NY, USA, 2015.

[191] Ganit Richter, Daphne R Raban, and Sheizaf Rafaeli. Studying gamification: the effect of rewards and incentives on motivation. In *Gamification in education and business*, pages 21–46. Springer, New York, NY, USA, 2015.

[192] Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. In *Fifth International AAAI Conference on Weblogs and Social Media*, pages 321–328, Menlo Park, CA, USA, 2011. AAAI.

[193] Leah R Rosenzweig, Bence Bago, Adam J Berinsky, and David G Rand. Happiness and surprise are associated with worse truth discernment of covid-19 headlines among social media users in nigeria. *Harvard Kennedy School Misinformation Review*, 1:1–37, 2021.

[194] Richard M Ryan and Edward L Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1):68, 2000.

[195] Niloufar Salehi and Michael S Bernstein. Ink: Increasing worker agency to reduce friction in hiring crowd workers. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 25(2):1–17, 2018.

[196] Niloufar Salehi, Lilly C Irani, Michael S Bernstein, Ali Alkhatib, Eva Ogbe, Kristy Milland, et al. We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In *Proceedings of the 33rd annual ACM Conference on Human Factors in Computing Systems*, pages 1621–1630, New York, NY, USA, 2015. ACM.

[197] Elliot Salisbury, Ece Kamar, and Meredith Ringel Morris. Toward scalable social alt text: Conversational crowdsourcing as a tool for refining vision-to-language technology for the blind. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*, HCOMP, pages 147–156, Menlo Park, CA, USA, 2017. AAAI.

[198] Ugo Braga Sangiorgi, François Beuvens, and Jean Vanderdonckt. User interface design by collaborative sketching. In *Proceedings of the Designing Interactive Systems Conference*, pages 378–387, New York, NY, USA, 2012. ACM.

[199] Saiph Savage, Chun Wei Chiang, Susumu Saito, Carlos Toxtli, and Jeffrey Bigham. Becoming the super turker: Increasing wages via a strategy from high earning workers. In *Proceedings of The Web Conference 2020*, pages 1241–1252, 2020.

[200] Saiph Savage, Andres Monroy-Hernandez, and Tobias Höllerer. Botivist: Calling volunteers to action using online bots. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 813–822, New York, NY, USA, 2016. ACM.

[201] Ben Schmidt. Ranking cs graduate programs, 2020. Accessed on December 20, 2022.

[202] Linda See, Alexis Comber, Carl Salk, Steffen Fritz, Marijn van der Velde, Christoph Perger, Christian Schill, Ian McCallum, Florian Kraxner, and Michael Obersteiner. Comparing the quality of crowdsourced data contributed by expert and non-experts. *PloS one*, 8(7):e69958, 2013.

[203] Stefan Seidel, Felix Müller-Wienbergen, and Jörg Becker. The concept of creativity in the information systems discipline: Past, present, and prospects. *Cais*, 27(2):14, 2010.

[204] Jami J Shah, Noe Vargas-Hernandez, Joshua D Summers, and Santosh Kulkarni. Collaborative sketching (c-sketch)-an idea generation technique for engineering design. *The Journal of Creative Behavior*, 35(3):168–198, 2001.

[205] Pao Siangliulue, Joel Chan, Krzysztof Z Gajos, and Steven P Dow. Providing timely examples improves the quantity and quality of generated ideas. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, pages 83–92, New York, NY, USA, 2015. ACM.

[206] Ángel Obregón Sierra and Jorge Oceja Castanedo. University students in the educational field and wikipedia vandalism. In *Proceedings of the 14th International Symposium on Open Collaboration*, pages 1–7, 2018.

[207] J Sinclair, Paul J Taylor, and Sarah Jane Hobbs. Alpha level adjustments for multiple dependent variable analyses and their applicability–a review. *International Journal of Sports Science and Engineering*, 7(1):17–20, 2013.

[208] Anjali Singh, Christopher Brooks, and Shayan Doroudi. Learnersourcing in theory and practice: Synthesizing the literature and charting the future. In *Proceedings of the Ninth ACM Conference on Learning@ Scale*, pages 234–245, New York, NY, USA, 2022. ACM.

[209] C Estelle Smith, Bowen Yu, Anjali Srivastava, Aaron Halfaker, Loren Terveen, and Haiyi Zhu. Keeping community in the loop: Understanding wikipedia stakeholder values for machine learning-based systems. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, New York, NY, USA, 2020. ACM.

[210] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[211] Anselm Strauss and Juliet Corbin. *Basics of qualitative research*, volume 15. Sage Publications, Thousand Oaks, CA, USA, 1990.

[212] Gail M Sullivan and Anthony R Artino Jr. Analyzing and interpreting data from likert-type scales. *Journal of graduate medical education*, 5(4):541–542, 2013.

[213] Lingyun Sun, Wei Xiang, Shi Chen, and Zhiyuan Yang. Collaborative sketching in crowdsourcing design: a new method for idea generation. *International Journal of Technology and Design Education*, 25(3):409–427, 2015.

[214] Pedro A Szekely, Daniel Garijo, Jay Pujara, Divij Bhatia, and Jiasheng Wu. T2wml: A cell-based language to map tables into wikidata records. In *ISWC Satellites*, pages 45–48, New York, NY, USA, 2019. ACM.

[215] Scott Thacker and Mark D Griffiths. An exploratory study of trolling in online video gaming. *International Journal of Cyber Behavior, Psychology and Learning (IJCBPL)*, 2(4):17–33, 2012.

[216] David Thissen, Lynne Steinberg, and Daniel Kuang. Quick and easy implementation of the benjamini-hochberg procedure for controlling the false positive rate in multiple comparisons. *Journal of educational and behavioral statistics*, 27(1):77–83, 2002.

[217] Todd M Thrash, Laura A Maruskin, Scott E Cassidy, James W Fryer, and Richard M Ryan. Mediating between the muse and the masses: Inspiration and the actualization of creative ideas. *Journal of personality and social psychology*, 98(3):469, 2010.

[218] Carlos Toxtli, Siddharth Suri, and Saiph Savage. Quantifying the invisible labor in crowd work. *Proceedings of the ACM on human-computer interaction*, 5(CSCW2):1–26, 2021.

[219] Christopher W Tyler and Lora T Likova. The role of the visual arts in enhancing the learning process. *Frontiers in human neuroscience*, 6:8, 2012.

[220] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledge base. *Communications of the ACM*, 57(10):78–85, 2014.

[221] W3C. Model for tabular data and metadata on the web. https://www.w3.org/TR/tabular-data-model/, 2015. [Online; accessed 2020-04-20].

[222] Shaun Wallace, Zoya Bylinskii, Jonathan Dobres, Bernard Kerr, Sam Berlow, Rick Treitman, Nirmal Kumawat, Kathleen Arpin, Dave B Miller, Jeff Huang, et al. Towards individuated reading experiences: Different fonts increase reading speed for different individuals. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 29(4):1–56, 2022.

[223] Shaun Wallace, Brendan Le, Luis A Leiva, Aman Haq, Ari Kintisch, Gabrielle Bufrem, Linda Chang, and Jeff Huang. Sketchy: Drawing inspiration from the crowd. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–27, 2020.

[224] Shaun Wallace, Alexandra Papoutsaki, Neilly H Tan, Hua Guo, and Jeff Huang. Case studies on the motivation and performance of contributors who verify and maintain in-flux tabular datasets. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–25, 2021.

[225] Shaun Wallace, Lucy Van Kleunen, Marianne Aubin-Le Quere, Abraham Peterkin, Yirui Huang, and Jeff Huang. Drafty: Enlisting users to be editors who maintain structured data. In *Proceedings of the Fifth AAAI Conference on Human Computation and Crowdsourcing*, HCOMP, pages 187–196, Menlo Park, CA, USA, 2017. AAAI.

[226] Greg Walsh and Eric Wronsky. Ai+ co-design: Developing a novel computer-supported approach to inclusive design. In *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing*, pages 408–412, New York, NY, USA, 2019. ACM.

[227] Hao-Chuan Wang, Dan Cosley, and Susan R Fussell. Idea expander: supporting group brainstorming with conversationally triggered visual thinking stimuli. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pages 103–106, New York, NY, USA, 2010. ACM.

[228] Jing Wang, Gen Li, and Kai-Lung Hui. Do monetary incentives create a spillover effect on free knowledge contribution? evidence from a natural experiment. *Evidence from a Natural Experiment (June 25, 2018)*, pages 1–24, 2018.

[229] Kai Wang and Jeffrey V Nickerson. A literature review on individual creativity support systems. *Computers in Human Behavior*, 74:139–151, 2017.

[230] Andrea Wiggins, Greg Newman, Robert D Stevenson, and Kevin Crowston. Mechanisms for data quality and validation in citizen science. In *e-Science Workshops (eScienceW), 2011 IEEE Seventh International Conference on*, pages 14–19, Piscataway, NJ, USA, 2011. IEEE.

[231] Dennis M Wilkinson and Bernardo A Huberman. Assessing the value of coooperation in wikipedia. *arXiv preprint cs/0702140*, pages 1–14, 2007.

[232] Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. Axis: Generating explanations at scale with learnersourcing and machine learning. In *Proceedings of the Third (2016) ACM Conference on Learning@Scale*, pages 379–388, New York, NY, USA, 2016. ACM.

[233] Blake Williford. Sketchtivity: Improving creativity by learning sketching with an intelligent tutoring system. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*, pages 477–483, New York, NY, USA, 2017. ACM.

[234] Wei Xiang, Ling-yun Sun, Wei-tao You, and Chang-yuan Yang. Crowdsourcing intelligent design. *Frontiers of Information Technology & Electronic Engineering*, 19(1):126–138, 2018.

[235] Chengyuan Xu, Kuo-Chin Lien, and Tobias Höllerer. Comparing zealous and restrained ai recommendations in a real-world human-ai collaboration task. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2023.

[236] Ying Xu and Carleen Maitland. Participatory data collection and management in low-resource contexts: A field trial with urban refugees. In *Proceedings of the Tenth International Conference on Information and Communication Technologies and Development*, ICTD '19, pages 18:1–18:12, New York, NY, USA, 2019. ACM.

[237] Khushnood Z. Naqshbandi, Silas Taylor, Ajit G. Pillai, and Naseem Ahmadpour. Labour of love: Volunteer perceptions on building relatedness in online volunteering communities. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–6, New York, NY, USA, 2021. ACM.

[238] Omar F Zaidan and Chris Callison-Burch. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, HLT '11, pages 1220–1229, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[239] Shuo Zhang and Krisztian Balog. Web table extraction, retrieval, and augmentation: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(2):1–35, 2020.

[240] Shuo Zhang, Zhuyun Dai, Krisztian Balog, and Jamie Callan. Summarizing and exploring tabular data in conversational search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1537–1540, 2020.

[241] Xuefeng Zhang, Enjun Xia, Chao Shen, and Jiafu Su. Factors influencing solvers' behaviors in knowledge-intensive crowdsourcing: A systematic literature review. *Journal of Theoretical and Applied Electronic Commerce Research*, 17(4):1297–1319, 2022.

[242] Qian Zhao, Zihong Huang, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. Precision crowdsourcing: closing the loop to turn information consumers into information contributors. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1615–1625, New York, NY, USA, 2016. ACM.

[243] Zhe Zhao, Zhiyuan Cheng, Lichan Hong, and Ed H Chi. Improving user topic interest profiles by behavior factorization. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1406–1416, New York, NY, USA, 2015. ACM.

[244] Zhenpeng Zhao, Sriram Karthik Badam, Senthil Chandrasegaran, Deok Gun Park, Niklas LE Elmqvist, Lorraine Kisselburgh, and Karthik Ramani. skwiki: a multimedia sketching system for collaborative creativity. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 1235–1244, New York, NY, USA, 2014. ACM.

[245] Yong Zheng. Context-aware collaborative filtering using context similarity: an empirical comparison. *Information*, 13(1):42, 2022.

[246] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proc. VLDB Endowment*, 10(5):541–552, 2017.