MANAGING PERSONAL INFORMATION

WITH PRIVATE, ACCOUNTABLE CROWDSOURCING

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Petros Nicolas Kokkalis

June 2013

Abstract

Crowd-powered systems combine the power of human judgment and creativity with the speed and precision of computers. These systems can efficiently help people be more productive in ways that no single human assistant or computer program could accomplish alone.

To provide help with personal information management, crowd-workers need access to people's personal information. However, people are understandably reluctant to share their entire private dataset with online workers. I introduce privacy and accountability techniques for parsimoniously sharing private data with online workers and provide experimental evidence that people can be more productive with assistance from the crowd. These techniques develop crowdsourcing as a platform trustworthy and responsive enough to be integrated into personal information management.

This thesis develops these ideas through two crowd-powered systems. The first, TaskGenies, is a task list that automatically breaks down users' tasks into actionable steps that can be completed one at a time. These action plans are created through crowdsourcing and reused through natural language processing when possible. The second system, EmailValet, is a web-based email client that introduces valet crowdsourcing. With EmailValet, users can share a limited subset of their inbox with online human assistants who extract embedded tasks from these emails. The system mediates and logs assistant access to establish accountability.

These systems point to a future in which people are personally empowered by the crowd and people's private data can be entrusted to crowdsourcing.

Acknowledgements

I thank Scott R. Klemmer for introducing me to HCI research and tirelessly advising the work of this dissertation; Mendel Rosenblum for his patient advice and generous support during my entire Ph.D. student career; Michael S. Bernstein for his candid advise and help on the second half of this thesis; Monica Lam for her support in recruiting participants to experiments and for being in my reading committee; Terry Winograd for the mind-opening conversations; Cliff Nass for being the chair of my oral committee; Thomas Köhn, Johannes Huebner, Dima Chornyi, Carl Pfeiffer, Moontae Lee, Florian Schulze for collaborating with me in the papers and the systems of this dissertation; Steven Diamond, Michael Chang, Dominic R. Becker, Binna Kim, Ryan Globus, Thomas Bridges-Lyman, and Arun Prasad for their contributions during their summer internships; Odysseas Tsatalos, Greg Little, Wendy Mackay, Cameron Teitelman for the thoughtful discussions; my Master's thesis advisor Vassos Hadzilacos for introducing me to theoretical computer science research and for recommending me when I applied to Stanford; my Bachelor's thesis advisor Manolis Katevenis for teaching me computer architecture and for being confident that I could earn a PhD at a top university; oDesk for sponsoring crowd workers; the StartX staff

for participating in many of the pre-studies of this thesis and bearing with premature software; the participants of these studies; the crowd workers and assistants who made the studies possible; Chengdiao Fan for the intelligent conversations and patient support through the ups and downs of my Ph.D.; my parents Goody and Kostas and my sister Teresa for their love and encouragement; and my grandparents who I promised to be a doctor by the time we meet again. This research was sponsored in part by NSF POMI 2020 Grant No. CNS-0832820.

Table of Contents

List of Illustrations

Chapter 1

Introduction

This dissertation hypothesizes that crowdsourcing can enhance personal information management (PIM) by organizing work and lowering activation energy for activities that people are either unwilling or don't have time to do themselves while also maintaining privacy and accountability.

Crowdsourcing is the process of utilizing the work of a large or elastic group of people [75]. The *wisdom of the crowd* [86] sometimes can accomplish things that neither individuals nor computers can do. For tasks that an individual is capable of but unwilling or doesn't have the time to do, crowdsourcing offers an elastic labor pool to accomplish those tasks [75]. The wisdom of the crowd is clearly demonstrated in the classic Galton's ox study, in which approximately 800 farmers were asked to assess the weight of a specific ox, and the average of all of the farmers' estimates was more accurate than any individual's estimation [33]. Similar results were observed in a very different domain where students were asked to guess the number of jellybeans in a jar

[88]. In both cases, effective crowdsourcing requires independent voting. The last decade has seen an explosion of online crowdsourcing that intelligently integrates the estimates and work of multiple independent workers to solve complex problems for which mere averaging would be insufficient. This online crowdsourcing has many useful applications, including creating the world's largest encyclopedia [60], labeling images [25], improving writing [13], helping blind people read [15], and analyzing protein folding [36]. To date, paid crowdsourcing applications and research have focused on tasks that require little accountability, expertise, or context.

Email and task management are key PIM domains and the focus of this thesis. Both problems require planning, making decisions (for example, deciding task priorities), and developing implementation intentions (i.e., creating action plans) [95]. All of these activities consume willpower [8], can be exhausting and boring [3], and cause ego depletion [95]. As a result, people often postpone making planning decisions or avoid planning completely, leading to unmet goals or missed deadlines [3]. Privacy issues make crowd-powered management of personal or sensitive information challenging. People may feel uncomfortable giving online workers unrestricted access to their personal and contextual information. Existing crowdsourcing methods offer little protection around privacy for applications that require a high level of accountability and contextual knowledge about the individual.

This thesis provides experimental evidence that crowdsourcing can effectively manage personal information, and introduces techniques for addressing user concerns around privacy and accountability. Two problems of personal information management drove [18] this research: task management and email overload. Both

aspects involve deeply personal information and the crowdsourcing literature contains little previous work on these topics. This thesis presents innovations in crowdsourcing developed to solve these driving problems. First, the *Genies* pattern (Chapter 3) strategically presents examples of similar work in the middle of a worker's thinking process to bubble up good ideas without constraining creativity and innovation. The Genies pattern also crystallizes the notion of efficiently using the crowd only for novel solutions and algorithmically re-applying existing solutions when possible. Individual solutions effectively become templates that can be re-applied to many instances. Second, the *Valet* approach (Chapter 4) introduces privacy and accountability mechanisms to enable crowdsourcing of applications with private or sensitive information. Valet crowdsourcing has the potential to fundamentally change PIM. We envision a future where we no longer need to individually manage our personal information.

This dissertation's primary behavioral finding is that people don't need to create the PIM planning and organizing themselves to get the benefit from it – externally-created plans are also beneficial. The interaction design and user experience finding is that crowdsourcing can be the engine for that external creation. Valet crowdsourcing can simplify aspects of personal organization that people are not willing to do themselves, such as breaking down each of their tasks into actionable steps and maintaining a constantly-up-to-date task list. Prior research has shown that developing implementation intentions [67] helps people complete tasks, but people are unwilling to do this upfront planning [3]. Chapter 3 shows that the crowd, without the efforts of the task owner, can do this task decomposition effectively. Chapter 4

provides experimental evidence that creating and maintaining a person's up-to-date task list can be successfully performed by external crowd assistants.

To enable this research, we built the TaskGenies and the EmailValet systems, representing more than 9 man-years of work[1], spread across 17 people. The systems have been available online over a period of three years. During this time, TaskGenies produced more than 20,000 action plans for people and EmailValet was featured in TV news stories, newspapers, and online publications that tagged it as disruptive and innovative.

Looking forward, there are many unanswered questions. How can knowledge about an individual can be institutionalized in crowdsourced systems to provide high quality 24-hour support? How can we guarantee fair payment to crowd workers? How can crowdsourcing be applied to more complex verticals, such as medicine, law, education, design, or even research itself? How will the labor market scale? How should the practical challenges like the systems and practice difficulties of tinkering with someone's personal information be addressed?

1.1 The Crowd Helping People Plan Tasks

This dissertation first takes up the challenge of harnessing the power of the crowd to improve task management. People complete tasks faster when they develop

[1] 9 man years = 3 years by the author of this dissertation + 6 months by 6 fulltime Master's students + 3 months by 3 fulltime Master's students + 4 months by 7 undergraduate interns. This calculation does not include the invaluable time of the faculty advisors.

concrete implementation intentions [3, 5, 59, 35, 62, 67]. Several experiments have found that people assigned to make concrete plans – from getting flu shots [67] to exercising for heart attack recovery [62] – follow through more often than those only required to formulate a high-level goal. This follow-through benefit could arise from the availability of an action plan (regardless of its source) and/or the process of contemplating a plan oneself. This work seeks to disambiguate these possibilities.

This thesis introduces and evaluates crowdsourcing and community approaches for creating plans, and NLP techniques for reusing them. A between-subjects experiment (n=280) found that people receiving crowd-created plans completed about 20% more tasks than those in a group prompted to create their own plans and those in a control group that did not create plans. Crowd-created action plans were especially effective for lingering and high-level tasks. To scale plan provisioning, a second experiment (n=388) assessed the efficacy of community-provided plans, finding them beneficial to participants. To further increase scale, we introduce and evaluate an NLP technique for blending and bootstrapping crowdsourced and automated results. To enable these experiments, we created TaskGenies: a crowd-powered task management system. This work introduces the Genies workflow, which combines the benefits of crowd wisdom, collaborative refinement, and automation.

The TaskGenies system provides custom, crowd-powered action plans for tasks that linger on a user's to-do list, as in Figure 1. The system is open to the public and has produced over 21,000 action plans.

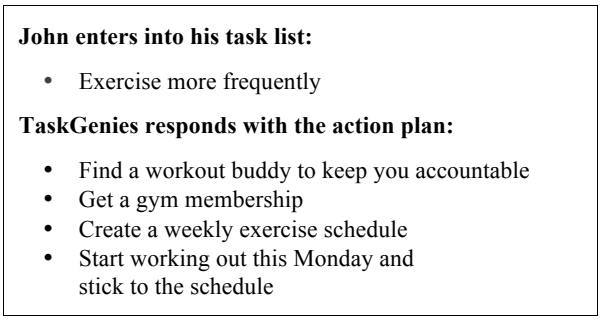

Figure 1 Decomposing people's tasks to concrete steps (action plans) makes high-level tasks more actionable. This way, tasks linger less and people complete more of them. Online crowds create new plans; algorithms identify and reuse existing ones.

To help workers efficiently produce good plans, TaskGenies strategically shows them examples of prior work. The Genies approach seeks to address the tension created by the contradictory effects of using examples. On one hand, viewing others' high-quality work can increase performance by norming expectations to a high level [6], and on the other hand, viewing existing answers to a current task also risks lazy copying and/or priming-induced conformity [84].

To balance this aforementioned tension, the Genies workflow employs prior work examples in two different ways. First, Genies workers initially rate solutions to related but different problems. Upfront rating helps convey norms, strategies, and the expectation of peer assessment. Second, midway through solving the problem, Genies makes others' solutions to the current problem available for workers to view, adapt, and integrate.

1.2 Private, Accountable Crowdsourcing

Email management means triaging a never-ending tide of incoming requests [63]. New messages push important requests out of view, which sometimes causes those requests being unintentionally missed [63, 99]. To prevent overlooking

important messages, people spend large amounts of time carefully processing their inbox or triage by focusing only on high priority messages [7, 31, 89]. However, people often keep unfinished tasks in their inbox [99] and triaging is error-prone [89]. As a result, tasks are mixed with other emails, get pushed down the stream by new messages, become hard to find, and are eventually lost.

Current approaches for handling email-based tasks are limited and/or expensive. Integrating task management directly into the email client [22,11] or asking communicators to structure their requests [100] requires significant manual effort, that in practice many people don't do. Automatic techniques have shown some promise in identifying tasks in emails [20, 28, 32, 55] but they are not yet fully reliable [55] and require heavy-handed user interaction [20] and training [32]. Finally, while privileged individuals have long relied on personal assistants to help with email, their integration into the email client is limited (e.g. they fall back on sharing a password) and the cost can be prohibitive.

This work combines the accuracy and oversight advantages of personal assistants with the large-scale availability and affordability of crowdsourced experts. We recruit workers from expert crowdsourcing platforms to help extract tasks and share these assistants across multiple people. This multiplexing increases employment for assistants and affordability for users. To present these task-organized messages, this thesis introduces the EmailValet mail client. EmailValet's task list (Figure 2) condenses an inbox into actionable items, makes tasks more prominent and easier to track, increases efficiency through task-oriented interactions (rather than coopting

general email primitives, like marking as unread), and focuses attention on the most important emails.

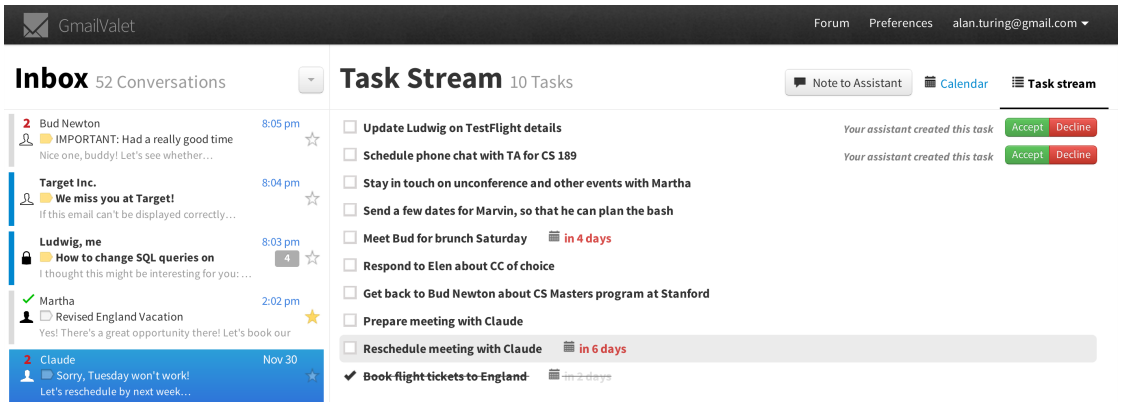

Figure 2. The EmailValet email client draws on crowdsourced expert assistants to transform a cluttered inbox into an organized task stream. Assistants are given limited, accountable access to the user's inbox so that they may extract tasks from each email.

To explore the potential for crowdsourced solutions, we conducted a formative survey finding that people feel a tension about recruiting remote assistants for managing complex, personal information: they want help, but they have reasonable concerns about giving strangers unfettered access. To address this tension, this thesis introduces the valet crowdsourcing approach. Like a valet key, valet interfaces seek a dual objective of parsimony and transparency. First, they parsimoniously give assistants just enough access to help with most of the work. Second, they make access boundaries transparent so users have an accurate model of what the assistant can and cannot do, and they cause transgressions to leave fingerprints. Achieving these dual objectives provides peace-of-mind for users and limits liability for assistants.

EmailValet illustrates this approach. For parsimony, EmailValet's access control shares a limited window of the inbox (by default, the 100 most recent

messages), and limits assistants' actions (to creating tasks). Users can also author a whitelist and blacklist of messages to be shared. For transparency, the user inbox is enhanced with icons that identify the messages that assistants accessed and the number of tasks each message contains. Furthermore, EmailValet provides a viewable log of all assistants' actions.

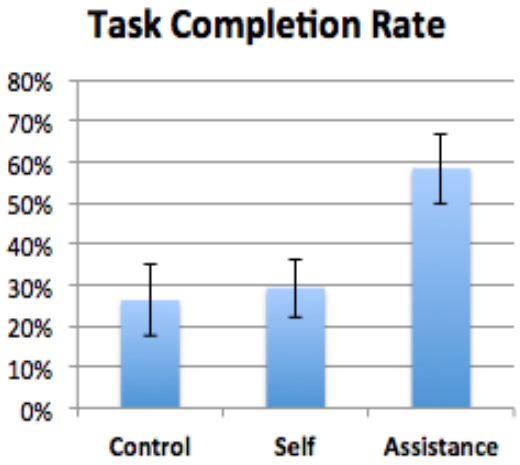

Figure 3. Participant task completion rates were significantly higher when assistants auto-extracted tasks for them.

This thesis hypothesizes that EmailValet helps people manage the tasks in their inbox, while maintaining acceptable privacy and accountability practices. A within-subjects experiment (n=28) compared participants' task completion rates when they only had standard email (Control condition) to when they extracted tasks themselves (Self condition) and to when they took advantage of crowdsourced expert task extraction (Assistance condition). With crowdsourced task assistance, people completed twice as many email-based tasks than with either standard email or self-extracting tasks with EmailValet (Figure 3). Participants grew more comfortable sharing their inbox with assistants as the study progressed.

1.3 Contributions of This Work

This dissertation provides the following crowdsourcing patterns and systems contributions.

- Genies crowdsourcing, which begins with independent ideation then shares examples for collaborative refinement.
- Valet crowdsourcing, like a valet key, grants limited access to private data and provides accountability.
- The TaskGenies and EmailValet PIM web applications help users manage tasks and email through crowd, community, and NLP assistance. These systems represent the first use of crowdsourced assistants for PIM.

These patterns and systems enabled the following behavioral contributions.

- Automatically provided action plans help people complete tasks.
- Crowdsourced assistants can manage personal information accurately, enabling EmailValet users to accomplish more tasks

Chapter 2

Related Work

This work draws from extant planning and email overload literature. It also builds upon crowdsourcing literature. Galton's ox [33] is an example of the wisdom of the crowd where averaging many independent estimates can attain a level of accuracy that no individual could. Similar to Find-Fix-Verify [13], Genies is a pattern that intelligently combines massive online work to solve problems where mere averaging is insufficient. The Valet pattern addresses privacy and accountability concerns while using online, elastic labor pools. The Valet pattern is orthogonal to the way worker contributions are combined or the number of workers used. To test the Valet pattern in isolation, EmailValet employed a single worker for each user in the study of Chapter 4. However, EmailValet is capable of having multiple workers per user to draw on multiple opinions and to provide 24-hour support.

This Chapter places this dissertation in the context of related research. The chapter begins with an overview of the planning literature. It continues with

describing crowd, community and algorithmic approaches for organizing crowd work. The chapter concludes with an overview of related work in email overload.

2.1 What Makes Action Plans Effective?

Popular and academic writing on work emphasizes that formulating actionable steps benefits both efficiency [3] and morale [5]. In this view, the key requirement for action is that steps be concrete. People are especially prone to ignore or procrastinate on creative, open-ended tasks because their intrinsic ambiguity requires selecting from multiple alternative implementations [72].

Moreover, having concrete, separated steps provides people with helpful guides on when to suspend work, especially to handle interruptions [41]. Limiting interruptions to subtask boundaries helps people complete work more quickly because it reduces the amount of state that needs to be recalled when resuming work [40,79]. Despite these benefits, people often fail to plan because the costs are immediate but the benefits are deferred [3,11].

With complex activities, the articulation work of planning and organizing tasks comprises an important piece of the job [10,85]. It may be that to benefit from an action plan one may need to create it oneself. There are several reasons why this might be the case. Sometimes, the benefit of action plan creation may reside in actively thinking through a plan. If mental simulation is the key ingredient, externally provided plans may have limited benefit. Moreover, the way people record tasks may require contextual knowledge for others to understand them. Can someone who lacks the

context of the person who needs to complete a task provide a useful plan? Finally, not all tasks may need plans. Small, actionable tasks may not benefit from decomposition.

2.2 Approaches for Organizing Crowd Work

Crowdsourcing is most valuable when algorithms alone can't provide sufficient results [91]. However, the cost and difficulty of acquiring labor presents a bottleneck to broader crowdsourcing use. This section summarizes prior work on the three major strategies for plan provisioning: crowd creation, community creation and algorithmic reuse.

Crowd Creation

The literature offers several approaches for organizing crowd workers to solve open-ended problems [49, 56]. The simplest way to allocate workers is to assign exactly one worker per task [61]. However, not all crowd workers are diligent and/or competent [61].

To increase quality, multiple workers can be redundantly assigned the same task. Figure 4 sketches alternative approaches. Worker solutions can be averaged, computed by pairwise agreement [91] or simple majority vote [90, 92], or weighted by estimated worker quality [23, 45]. This independent redundancy parallelizes naturally and, under certain conditions, yields higher quality [86].

The number of workers to assign to tasks can even be decided adaptively (e.g., [23]). For more sophisticated tasks, workers can collaboratively refine others' work (e.g., [61,103]) or be assigned different roles (e.g., [13,48,54,61]). Some workflows

first divide work into smaller subtasks, then have a group of workers do the work, and finally let another group of workers assess the results (e.g., CastingWords.com, [13], [57]). More complex approaches delineate a separate phase for merging and refining different ideas [2,48]. Other techniques provide workers with more flexible workflow control to choose what they do (do work, refine, assess) [103]. The appropriateness and relative merits of these different approaches are being actively studied.

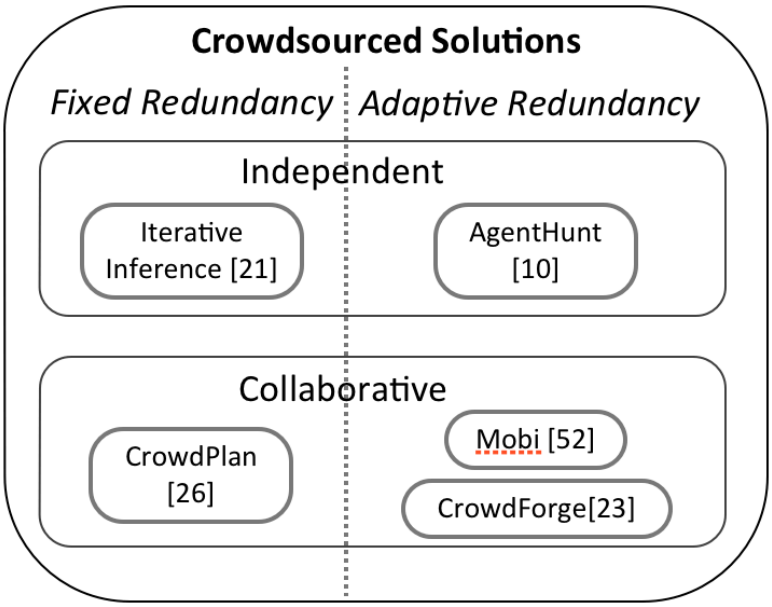

Figure 4 A sketch of different approaches for assigning workers to tasks, showing exemplars of key alternatives. Quality can be improved through redundancy, refinement, or hierarchical approaches using a fixed or an adaptive work redundancy.

Overall, collaborative approaches allow for the good ideas of one worker to propagate to the solutions of other workers, but they may suffer from worker fixation to existing ideas. Independent approaches allow for broad ideation and do not have this fixation problem but make diffusion of good ideas among workers impossible. Hybrid approaches, like Genies, try to combine the benefits of both approaches.

Closest in domain to this work, CrowdPlan [57] demonstrated the feasibility of having crowd workers decompose high-level mission statements (such as spending a day in a city) into concrete steps (such as visiting a museum). While workers' iterative refinement often yielded impressive results, the paper's examples also include flaws, like nearly identical steps, suggesting that iteratively incorporating results from multiple people may compromise coherence. Our work contributes empirical results showing the efficacy of crowd-created plans.

Techniques also differ in the way they calibrate workers. Some show examples in the beginning [103], others use them to indicate what not to produce [57], still others do not use any examples.

Community Creation

To recruit labor, it can be effective to leverage the user community and user's social networks [14]. Routing tasks to peers can be handled by self-declared expertise [1], by publicly posting to an online forum [1,66], by posting to one's social network [14,69], or automatically based on inferred expertise and availability [38]. When planners help each other, it helps systems scale and can engender a community ethos.

Automatic Reuse

Identifying and presenting relevant examples can enable content reuse. People can actively search for such examples [17], or they can be automatically provided [93]. Algorithmic approaches are appealing because additional users can be handled without additional human effort. However, algorithmic approaches are only appropriate when the relevant content already exists [101].

2.3 Email Overload

EmailValet extends work on email overload, automated email management, and task extraction.

Email overload is an enduring challenge [31, 99]. While email has many positive impacts on organizations [21], frequent email interruptions can decrease productivity and large volumes can create information overload [39, 42, 96].

Data mining, categorization, and metadata extraction can help users organize their email. These approaches ease email filing [19, 9, 16, 47, 80, 81, 83], detect important emails [19, 16, 37, 83], summarize messages [76], forward and reply to messages [39], and filter spam [78]. For example, Gmail's Priority Inbox identifies and highlights emails that the user is likely to want to read and respond to. Other work shares EmailValet's goal of detecting action items and tasks in emails [12, 20, 32, 83] or even automating their execution (*e.g.*, booking a meeting room) [32]. However, these machine-learning approaches lack human "common sense" and often suffer from false positives [20] that a professional assistant would not make [27]. Consequently, EmailValet reaches out to human assistants.

Information workers manage many of their tasks using their email clients [55,99]. Common mail clients (e.g., Outlook, Gmail) allow people to mark a message as a task and set a due date [22, 71]. Taskmaster grouped message threads by task and allowed users to customize these groupings [11]. TimeStore-TaskView is an email interface that is centered around the relationship between messages, with the focus on managing future tasks [37]. EmailValet, like TaskView and Taskmaster, foregrounds

tasks rather than messages. It differs from this prior work by integrating human assistants into the loop to lower users' organization burden. EmailValet's inbox interface also adds iconic cues that reflect assistants' activity.

Executive assistants help stem the tide of email overload, focusing their principal's attention on important messages, shielding them from unimportant ones, and handling simple tasks autonomously [27]. Reflecting this, mail clients such as Microsoft Outlook allow users to delegate limited inbox access to assistants [4, 82]. This work generalizes and extends these relationships to a large class of new users and assistants. To do so, we introduce techniques for an assistant to support multiple users and focus on building common ground and maintaining transparency with remote assistants. Assistants in the current deployment of EmailValet are drawn from an elastic labor pool and each works continuously with a number of users. This notion of crowdsourcing extends it to be more than just the micro-multiplexing approach demonstrated in Amazon Mechanical Turk. While the same functionality can be provided using Mechanical Turk workers, this dissertation opted to use a smaller worker pool to decouple the issues of organizing crowd work with privacy and accountability concerns.

We hypothesize that the manual effort of user-controlled systems and low accuracy of automated systems have yielded low adoption in practice. Our intuition was that paid crowd assistants could be a high-accuracy, low-effort, and affordable solution, but that user trust would be a main concern. Previous crowd-powered systems (e.g., [13]) have largely sidestepped the challenges of sharing private data with crowd workers.

Chapter 3

TaskGenies

People complete tasks more quickly when they have concrete plans. However, they often fail to create such action plans. (How) can systems provide these concrete steps automatically? This work[2] demonstrates that these benefits can be realized, not only when people plan for themselves, but also when these plans are created by others or reused from similar tasks. Three experiments test these approaches, finding that people indeed complete more tasks when they receive externally-created action plans. To automatically provide plans, this work introduces the Genies workflow that combines benefits of crowd wisdom, collaborative refinement, and automation. We demonstrate and evaluate this approach through the TaskGenies system, and introduce an NLP similarity algorithm for reusing plans. The studies in this work found that it is

[2] This work was also presented in [51]

possible for people to create action plans for others, and showed that this can be done cost effectively.

3.1 Hypotheses and overview of experiments

Four experiments evaluated the potential of providing action plans and the Genies approach. The following subsections elaborate the studies' rationale and hypotheses.

Auto-Provided Plans Increase Task Completion Rate

Can crowdsourced workers provide good action plans? This thesis hypothesizes that, yes, crowd-created action plans can be relevant, useful, and help people complete more tasks. We hypothesize that automatically provided plans can get people to an actionable state more frequently and with less effort than if they were left to their own devices. Action plans may also provide tactics or insight that people lack on their own.

Main Hypothesis: *Automatically providing action plans helps people complete more tasks.*

To evaluate the potential of externally created action plans, this work compared participants' task completion rates in three between-subjects experiments that collectively compare crowd-, co-, and self-production; recycling plans, and a control without explicit planning prompts.

Action Plans Differentially Benefit Different Task Types

When tasks are daunting, complex, and/or novel, an action plan may improve motivation or help define a clear path. However, crowd-created action plans might not always be beneficial. If a task is already small and well defined, there may be no advantage in dissecting it further. Furthermore, the crowds may be unable to help if a task is difficult to understand, vague, or requires a lot of contextual knowledge. When a plan is not needed or inaccurate, the system should make it easy to ignore.

Actionability Hypothesis: *Automatically providing action plans helps people more with high-level than with small, well-defined tasks.*

Procrastination Hypothesis: *Automatically providing action plans help people more with lingering tasks than with recently added tasks.*

Comprehension Hypothesis: *Plan creators are more effective when tasks are clearly worded and/or require minimal contextual knowledge.*

Study 1: Do Externally-created Action Plans Help?

A between-subjects experiment compared task completion rates for a Crowd group where anonymous crowd workers provided participants with action plans, a Control group without this action-plan support, and a Prompt group who were prompted to create action plans for themselves. The study evaluated participants' overall task completion and analyzed completion rates of different types of tasks.

The study found crowd-created plans significantly increase participants' completion rates compared to both the Prompt and Control conditions (see Figure 5). It also found no significant difference between Prompt and Control. Furthermore,

Crowd scored better than Control in every single observed type of task. The largest differences were on tasks that the task owner described as high-level and/or lingering.

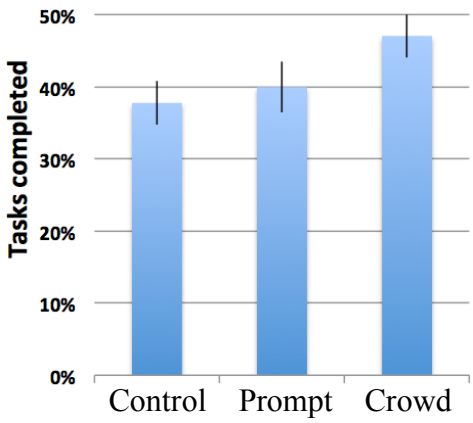

Figure 5 Participants in the Crowd completed significantly more tasks than those in the Control and Prompt conditions. Error bars indicate 95% CI.

Scaling with Community-Created Content

The cost of crowdsourcing rises in proportion to the supported user base. Furthermore, sufficient crowd labor may not always be available. How can we make such a system efficiently scale to many users? If users also contribute action plans, this lessens the dependence on a paid crowd. But will people be willing to contribute as much to the system as they expect to get out of it? Will they create the same level of quality plans as paid crowd workers? And will this have an influence on their own completion levels? Also, are community created plans helpful?

Community Hypothesis: *Community-created plans, like crowd-created plans, also help people complete more tasks.*

Study 2: Can Plans Be Sourced from a User Community?

A second study explored having the user community generate content as a scaling strategy. This between-subjects experiment compared participants who created action plans for each other (Community) against a Control group. (Because the first study found no significant difference between Prompt and Control, we simplified the design to two conditions.) This experiment measured task completion rate and contribution level.

The Community condition significantly outperformed the Control condition, but Community participants did not produce as many action plans as they expected to receive.

Action Plans Can Be Reused for Multiple People

Creating new action plans for every task seems wasteful and costly, especially when tasks are similar or repeated. Crowdsourcing platforms, such as Amazon Mechanical Turk, have limited throughput. Community-based solutions may not have a balanced capacity of work needed versus work produced. If reusing plans for similar tasks is helpful to people, then reuse can offer a significant scaling benefit. However, sufficiently similar tasks may not arise frequently and/or algorithmically identifying similarities may not lead to good results due to natural language obstacles.

This thesis hypothesizes that many tasks are sufficiently similar as to be usefully reused, and that NLP-based reuse is tractable.

Reusability Hypothesis*: The same action plan can help multiple people complete similar tasks.*

Study 3: Can Algorithms Enable Plan Reuse?

The third experiment investigated further workload reductions by algorithmically reusing existing action plans. For a Recycle condition, we designed an algorithm that selected an action plan based on the similarity of a task against a corpus of tasks with existing action plans. The task completion rates of participants of the Recycle condition were compared against a Control group.

The Recycle group completed significantly more tasks than the Control condition.

The results of the Community and Recycle experiments show how plan provisioning can scale for a large number of people.

Study 4: How Does Genies Compare to Simple Alternatives?

Was the sophistication of Genies necessary or could one achieve similar results with simpler alternatives? An experiment compared Genies to a serial, a parallel, and a revision workflow. Participants produced the best work when assigned the Genies workflow.

3.2 The TaskGenies System

The TaskGenies crowd-powered task-management community lets users manage their tasks through mobile and web interfaces, and create action plans for each other. We created the system to evaluate our hypotheses and refined it iteratively based on the findings. The system is open to the public and has produced over 21,000

action plans. The following subsections elaborate how tasks are managed in TaskGenies.

Multiple Convenient Ways to Enter Tasks

TaskGenies has three interfaces for submitting tasks. First, a user can send a task by email to tasks@taskgenies.com. Second, they can visit taskgenies.com/entry to submit a task they intend to do. Third, they can visit my.taskgenies.com, a to-do list Web application, to enter and manage tasks and receive action plans (Figure 6 and 7). Users can request plans for tasks, and have the option to auto-request plans when a task lingers for several days.

Figure 6 TaskGenies Web interface. Users can add steps of a provided action plan (right) as sub-tasks (left).

Note that study participants used a Web form rather than the TaskGenies to-do list interface to enter tasks, so they could not create task hierarchies.

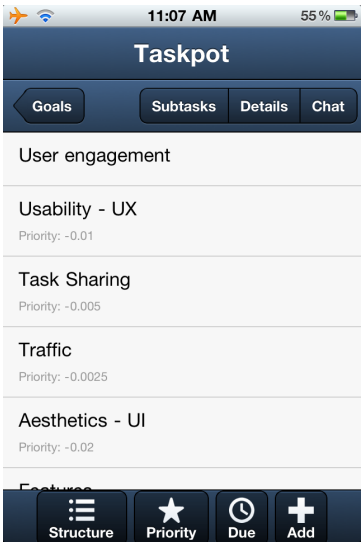

Figure 7 TaskGenies mobile task list.

Receive (New or Reused) Action Plans Automatically

To provide an action plan, the system first compares the given task with existing ones. If a similar task with an action plan is found, this plan is returned to the user. If there is no similar task in the database, the system crowdsources the creation of a new plan. Multiple different crowd workers on Amazon Mechanical Turk are employed to decompose each task into an action plan using TaskGenies. Once it is available, the system emails the plan to the respective user and displays it next to the corresponding tasks on the user's task list (see Figure 6). Users of my.tasksgenies.com can resubmit the task to the crowd, providing clarifications, if they don't like the action plan they received.

NLP Identifies Similar Tasks to Reuse Action Plans

The GeNiLP Natural Language Processing technique identifies similar tasks and reuses action plans. This technique, given a corpus of existing tasks C and a new

task t, outputs the most similar task s ∈ C and a similarity coefficient c ∈ [0,1]. The higher the c is, the more similar s is to t. The algorithm is trained on existing tasks with action plans. Appendix D summarizes its key features.

Often, task titles are fragments rather than complete or grammatical sentences. Consequently, GeNiLP treats tasks as a bag of words rather than as sentences. It uses WordNet's hierarchy [68] to match tasks with similar meaning (e.g., "buy pants" and "purchase trousers"; or "Email violin teacher" and "Call piano teacher to schedule lessons").

3.3 The Genies Crowdsourcing Pattern

This work introduces Genies, a novel crowdsourcing pattern that seeks to achieve the diversity benefits of multiple sources and the coherence benefits of individual creation. Each worker's task comprises three steps (Figure 8):

1. Rate: Taking a cue from calibrated peer assessment [77], Genies begins with a calibration step where workers view and rate results of different but analogous tasks. This serves several functions simultaneously: it finds good solutions through rating, trains and calibrates new workers, demonstrates that work will be assessed, and dissuades some lazy workers from continuing.

2. Contemplate: To encourage independent thinking, workers begin with a clean slate. In the middle of producing their own result, workers are shown others' results for the current or similar tasks, allowing them to integrate elements from prior examples when useful (see Figure 9).

3. Synthesize: The worker can draw inspiration from these examples and incorporate relevant ideas into their own work. This extracts original ideas from the current worker, bubbles up great existing ideas from previous workers, and hopefully converges on the best ideas. Examples can also provide conceptual help by demonstrating the space of possible alternatives [26,54,58,65].

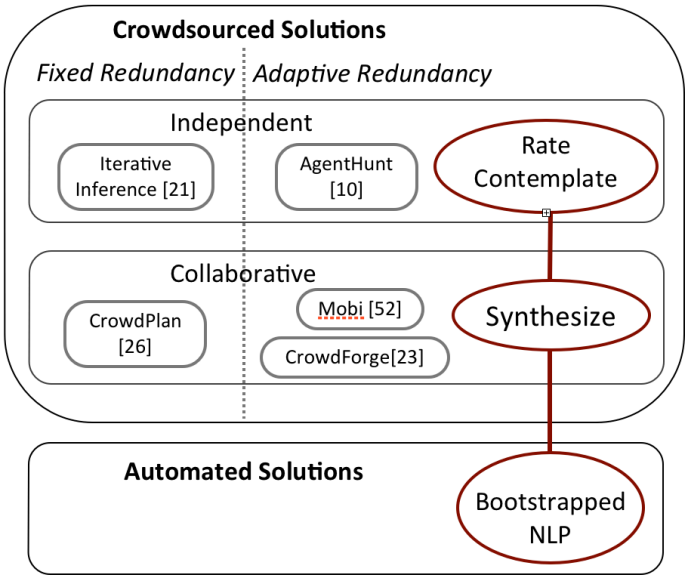

Figure 8 Genies crowdsources novel solutions. "Rating" and "contemplating" happen independently, whereas "synthesizing" collaboratively refines the final outcome. Existing solutions can be reused through NLP.

By making each worker responsible for a complete result Genies achieves higher coherence than more fragmented approaches. The quality of results depends on the number of ratings and the threshold required to accept a solution. Scaling efficiently requires high quality results with few workers. (The studies for this thesis alone generated 3,620 action plans.) Genies scales naturally by matching the number of upfront ratings each user performs to the number of ratings required for a decision.

TaskGenies required 5 ratings, converging on an accepted plan in an average of 2.2 workers per task.

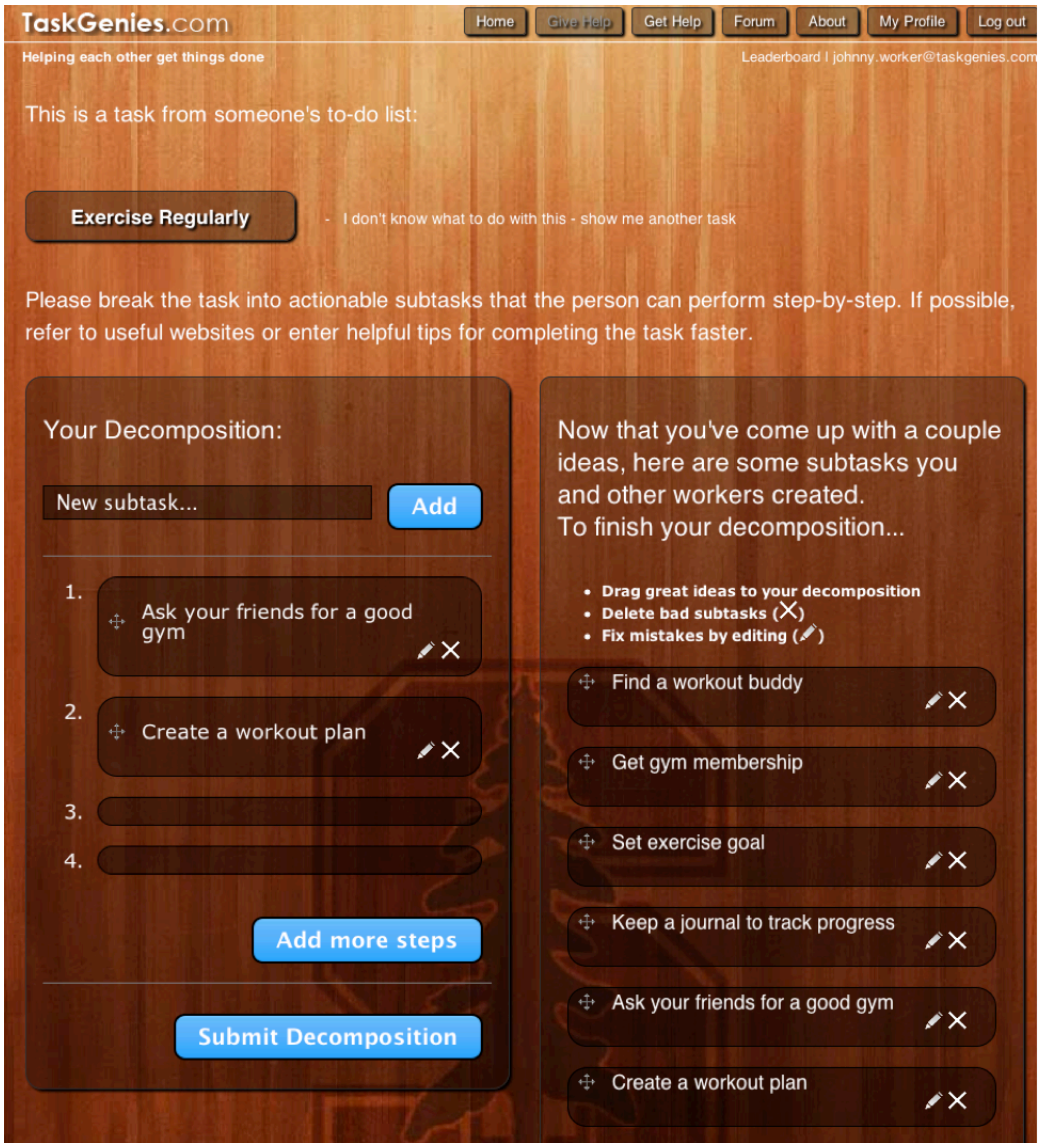

Figure 9 Workers use this interface to create action plans. After entering a couple of steps the system shows relevant examples on the right.

Applying Genies to Create Action Plans

In the rate step, TaskGenies presents workers with a Web page asking them to rate plans for other tasks. Rating assignments are randomized to discourage fraud. We

informally experimented with different numbers of ratings and average-rating thresholds for accepting a plan. Our experience is that averaging five ratings generally yielded good accuracy. TaskGenies selects the first action plan that receives a sufficiently positive average rating. What is "sufficiently" positive? We found that an acceptance threshold of 1.0 in a 5-point scale of -2 to 2 produces good results in an average of 2.2 HITs[3]. Increasing the threshold to 1.5 created better results but required several more HITs. Our experience was that thresholds above 1.8 risked extremely long running times without a commensurate boost in quality because very high average scores are rare for some tasks and workers. (Using the median might limit the negative impact of inaccurate workers.)

Relative to current results for iterative refinement (e.g., [57]), Genies uses fewer workers, but requires more work from each. On average, Genies provided TaskGenies with action plans using only 2.2 workers, balancing broad ideation and high quality without wasting labor. By contrast, CrowdPlan [57] used a larger number of smaller tasks, suggesting 10 workers for refinement and 5 for rating.

Next, a second page (Contemplate) presents workers with a new task, and requires them to enter two plan steps. Then, TaskGenies uses NLP to suggest steps that other workers created for similar tasks. These steps are ranked by usage popularity (see Figure 9). The worker is encouraged to review, drag-and-drop, and

[3] HIT stands for Human Intelligence Task. The term is used on Amazon Mechanical Turk to refer to one unit of work.

revise favorable suggestions (Synthesize). The system solicits workers to down-vote low-quality suggestions, improving future results. Workers may remove steps from view by pressing an "×" button; this lowers that step's ranking in the future by implicitly recording the deletion as a down-vote.

Crowd workers were encouraged to enter up to four steps per action plan (by displaying four empty input fields initially), but could use as many as they wished by creating additional steps (by clicking an add-button).

3.4 Study 1: Crowd-created Action Plans

Method

A between-subjects experiment compared the task completion rate of people who received emailed action plans from the crowd (Crowd condition), people who received email prompts to create their own action plans (Prompt condition), and those who were not explicitly asked to create plans (Control condition). Participants in all conditions were unaware of the TaskGenies Web site. Instead, participants submitted tasks through a generic Web form.

Participants

Two hundred eighty people, all U.S. residents and Internet users participated in this study: 13 reported 18-25 years old, 123 reported 26-35, 24 reported 46-55, 12 did not report age; 130 reported female, 147 reported male, 3 did not report gender. Participants were recruited through an online solicitation that offered participants a chance to win an iPad; one participant was randomly selected at the end of the study.

The study restricted crowd workers to people with HIT acceptance rate 80% or more and paid $0.04 per HIT.

Procedure

An online form instructed participants to provide 10 tasks they hoped to complete in the near future and 10 lingering tasks ("tasks that have been lingering on your to-do list for a while"). Participants were randomly assigned to the following three conditions.

The two main conditions were:

Control: Ninety-four participants received no encouragement to create action plans for the tasks they entered and were left to their own devices.

Crowd: Ninety-three participants received action plans for their tasks by email. To create these action plans, participants' tasks were posted to TaskGenies, where crowd workers used Genies to generate action plans. When a plan's average rating exceeded +1 on a -2 to +2 scale (5 total ratings needed), it emailed the plan to participants (Appendix A).

But, sending plans by email may serve as a reminder for the task itself. To control for the effect of the reminders, the study included a third condition:

Prompt: Ninety-three participants were sent emails asking them to create action plans for their own tasks. To make this condition parallel to the crowd condition, each email listed one task, asked participants to create an action plan for it, and suggested: "Before you close this email, could you possibly start this task by

completing its first applicable step right away?" To ensure that the number and timing of emails was consistent with the crowd condition, each participant was randomly paired with a participant in the Crowd condition. Whenever TaskGenies emailed that Crowd participant an action plan, TaskGenies also emailed the corresponding Prompt participant. The wording and structure of these emails was designed to be as similar as possible. See Appendix B for such an email. The experiment did not require participants to submit plans; as such we cannot report compliance. This encourage-without-demand approach was designed to maximize realism.

Dependent Measures

Overall Completion Rate: The study measured how many tasks participants completed in a week. One week after the study began, the system sent all participants an email with a link to a web page listing their 20 tasks. Participants were instructed to mark the tasks they completed. The study computed the percentage of tasks completed by participant as the overall completion rate. Eighty-two participants responded to this survey from the Control, 78 responded from the Crowd, and 74 responded from the Prompt condition.

Comparing Crowd to Control measures the difference between the approach introduced in this thesis and current practice. Comparing Crowd to Prompt measured the potential benefits of externally-provided plans versus asking people to plan themselves. Comparing Prompt and Control measured the potential benefits of explicit prompting.

High-Level Tasks Completed: To better understand the impact on different task types, a follow up survey asked the Control and Crowd participants who reported completion rates about the nature of their tasks. (For simplicity, we omitted the Prompt condition, as there was no significant difference between Prompt and Control in completion rates). The survey asked participants to categorize each of their tasks as "high-level" or "small & well-defined". Sixty-three Control participants responded; they categorized 34.9% of 1093 tasks as high-level. Fifty-eight Crowd participants responded; they categorized 34.8% of 980 tasks as high-level. The analysis compared the number of tasks completed across categories (high-level versus well-defined for both conditions).

Lingering Tasks Completed: The analysis included all tasks with completion information excluding tasks where participants did not respond about completion. Eighty-two participants provided completion information for 1640 tasks in the Control condition and 74 participants provided completion information for 1480 tasks in the Crowd condition. The analysis compared the number of tasks completed across categories (lingering versus not for both conditions).

Understandable Tasks Complete: For every task categorized as high-level or well-defined, three independent workers on Amazon Mechanical Turk answered two questions: Would you need more context about this task to be able to break it down into smaller steps? Do you find this task too vague? The analysis used the majority answer for each question, comparing the number of tasks completed across categories (needs context, doesn't need context for both conditions; too-vague, not-vague for both conditions).

Results

Overall Completion Rate

Participants in the Crowd condition completed significantly more tasks (47.1%) than participants in the Prompt (40.0%) or Control (37.8%). An analysis of variances was performed with participant condition as a factor and participant task completion rate as the dependent variable, finding a significant effect of condition ($F(2,225)=4.85$, $p<.01$). Follow-up pairwise t-tests with false discovery rate (FDR) correction found a significant difference between Crowd and Control ($t=3.07$, $p<.01$), and between Crowd and Prompt ($t=2.19$, $p<.05$). No difference was found between Prompt and Control ($t=0.69$, $p>.05$). The results are graphically presented in Figure 5. Figure 10 shows the task list for an example participant in each condition. Figure 11 shows an example of a crowd-created action plan.

High-Level Tasks

Providing action plans significantly increased task completion rates for high-level tasks, and marginally for well-defined tasks. When provided an action plan, participants completed 1.49 times more high-level tasks (44.3% of 341 in Crowd vs. 29.7% of 381 in Control; $\chi2=16.00$, $p<0.0001$). Participants completed 1.13 times more well-defined tasks with an action plan (44.9% of 639 in Crowd vs. 39.6% of 712 in Control; $\chi2=3.68$, $p=0.055$). This suggests that crowd-created action plans have a larger effect for high-level tasks than well-defined ones.

CONTROL	CROWD
Vacuuming **Cleaning the bathroom** **Cleaning the kitchen** ***Mail my father's death certificate to creditors*** Buy warm clothes for camping ***Buy other supplies for camping*** *Bake banana bread for husband's co-worker* Mop the floor Change the water filter **Wash the cat's litter box and change out the litter** Identified as lingering: *Clean out the closet* *Have a yard sale* Call to check the status of our tax refund Make a doctor's appointment *Put some things up on Craigslist* **Clean the oven** Sort through books to sell *Buy travel supplies for our trip to Texas* **Rent a carpet cleaner to clean the carpet** Wash winter clothes	***Prepare for a short trip from Michigan to Colorado.*** **Purchase back to school items for kids.** Close my swimming pool for the season. **Research kennels for dogs to stay at for 5 nights.** Find an arm specialist in Kalamazoo. **Get an oil change on my 1999 Toyota Camry.** Post items to sell on eBay. Register my son for fall soccer in Kalamazoo, MI. Register my daughter for swim team in Kalamazoo, MI. Identified as lingering: **Put a hold on mail for one week while I am on vacation.** Replace panels in basement. Open a savings account for my son. **Update my resume.** Research desktop computers to purchase. **Schedule a hair cut and color appointment.** Trim bushes in the front yard. **Schedule vet appointment for routine shots.** **Clean out my email inbox.** Get photos backed up on a CD from my computer. Go through closet and get rid of old clothes.
PROMPT	
Figure out my flight to LA **Find things to do for trip to Atlantic Beach, NC** **Pay Bills** **Pack for my vacation** Get a pedicure **Buy some kindle books for the beach** **Get my books back from Julie** **Do a fire drill for the ladies at work** Set up plans with Katie *Buy some fall clothes* Identified as lingering: Research other jobs Research grad school Get in touch with Brittney to make plans **Organize my recipe book** Make a dentist appointment Get new glasses Organize my closet Look it to a show in NYC Get a graduation present for Mom Take Kate & Julie to something fun; sisters only	

Figure 10 Each box shows one participant's tasks. Completed tasks are shown in bold. Tasks deemed as high-level by participants themselves are shown in italics. We selected participants with completion rates around mean.

Lingering Tasks

Providing action plans significantly increased task completion rates for both lingering and non-lingering tasks. When provided an action plan, participants

completed 1.33 times more lingering tasks (40.6% of 780 in Crowd vs. 30.5% of 820 in Control; $\chi 2=17.72$, $p<0.0001$). Participants completed 1.13 times more non-lingering tasks with an action plan (53.1% of 780 in Crowd vs. 46.8% of 820 in Control; $\chi 2=6.24$, $p<0.05$). This suggests that crowd-created plans have a larger effect for lingering tasks than non-lingering ones.

ACTION PLAN FOR: Post items to sell on ebay

- Log in on your eBay account and familiarize yourself with the fee structure and decide which features you want on your listings.
- Set out items to be listed.
- Photograph each item, preferably against a tablecloth or other background. Take photographs against different colors of backgrounds and choose the most appealing photos for the eBay listings.
- Before you write your listings, do eBay searches for the same or very similar items and write down potential keywords you wish to include in your eBay description.
- Locate appropriate shipping containers for each item or price what various shipping methods cost and requirements for packaging.
- Require the Buyer to pay shipping costs. You are permitted to charge for the time and materials to prepare the item for shipping in addition to the shipping charges from USPS or UPS or whoever.
- Write up descriptions that accurately describe the item and its condition. Do not make statements that are not truthful or overstate the condition of your item. This can result in negative feedback and disputes.
- Consider whether you are going to require the buyer to pay for insuring the item in transit.
- Proofread your eBay listings and make sure you have uploaded the correct pictures.
- Consider setting a "buy-it-now" price so you do not have to wait out the term of the auction.

Figure 11 Crowd-created action plans can provide valuable information. However, this does not guarantee completion. See more action plans on taskgenies.com

Comprehendible Tasks

The study measures comprehensibility in terms of whether a task requires additional context for interpretation or is vague. Providing an action plan significantly improved completion rate for tasks with sufficient context, (44.7% of 828 in Crowd vs. 36.1% of 893 in Control; $\chi 2=12.94$, $p<0.001$) but not for those needing more context (44.7% of 152 in Crowd vs. 36.5% of 200 in Control; $\chi 2=2.11$, $p=0.15$). Similarly, providing an action plan significantly improved completion rate for tasks

that rated "not vague" (44.8% of 900 in Crowd vs. 35.6% of 990 in Control; $\chi 2=16.33$, $p<0.0001$), but not for vague tasks (43.8% of 80 in Crowd vs. 41.7% of 103 in Control; $\chi 2=0.014$, $p=0.9$). Raters' "vague" label correlated moderately with owners' "high-level" label ($R=0.367$), showing evidence that high-level tasks of one person can be perceived as vague by others.

3.5 Study 2: Community-created Action Plans

Study 1 found that crowd-provided action plans help people complete more of their tasks. What would happen if we asked people to participate and create plans for their peers? Would community-created plans still be helpful? How much would people be willing to contribute? Study 2 investigates these questions. If successful, community creation also alleviates throughput bottlenecks on crowdsourcing platforms.

Method

Participants

Three hundred eighty-eight people, all U.S. residents and Internet users, participated in this study: 212 people 18-25 years old, 94 people 26-35 years old, 23 people 36-45 years old, 14 people 46-55 years old, 14 people 56 years old or older, 32 people who did not disclose their age; 157 female, 221 male. As in study 1, participants were recruited through an online solicitation with an iPad raffle as the incentive and workers were restricted to 80% HIT acceptance rate and paid $0.04 per HIT.

Procedure

As in study 1, participants provided 10 lingering and 10 non-lingering tasks they were planning to complete in the near future. Participants were assigned to one of two conditions: Control or Community.

Control: The same as in study 1. Three hundred participants were assigned to this condition.

Community: Eighty-eight participants were assigned to this condition. (We initially recruited 300. This number would have been difficult to support interactively and inexpensively. Consequently, we scaled back to 88. This scaling challenge inspired the reuse approach in study 3.) At the beginning of the study, participants' tasks were posted on the TaskGenies community. During the study, participants were periodically instructed to visit TaskGenies and create action plans for others. Many participants did not create enough plans for their peers, and for realism we did not enforce this. Mechanical Turk workers provided plans for the remaining tasks. This community + crowd approach enabled participants to receive a full set of action plans. As in study 1, when a task received a plan, the system emailed it to the relevant participant (Appendix C). Each email encouraged the recipient to visit TaskGenies and create plans for others. The rest of the method was the same as study 1.

Dependent Measures

Completion Rate: Like study 1, this study measured how many tasks participants completed in a week. One week after the study began, the system sent all

participants an email with a link to a Web page listing their 20 tasks. Participants were instructed to mark the tasks they completed.

Contribution Rate: This study also measured how many actions plans each participant created for other participants.

Results

Completion Rate

Participants in the Community condition completed more tasks (55.5%) than participants in the Control (49.9%) condition. A pairwise t-test found a significant effect between Community and Control ($t=2.18$, $p<0.05$).

Contribution Rate

The 88 Community participants created 655 action plans. Therefore, on average, each participant created 7.44 action plans (SD=12.3). Amazon Mechanical Turk workers created the remaining 12.6 action plans per participant. Community creation resulted in a 37% reduction of load of the crowd workers. Figure 12 depicts the distribution of contribution among participants.

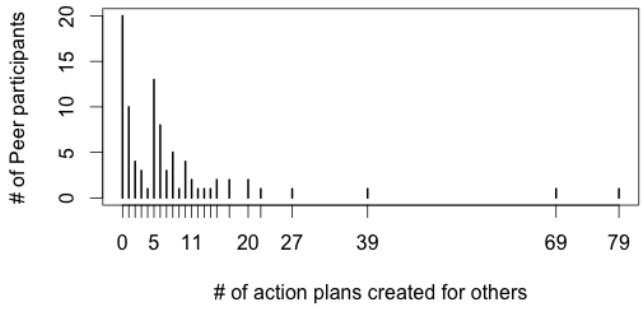

Figure 12 The study ensured that every Community participant received 20 action plans, augmenting with paid and volunteer work. However, participants contributed at very different levels.

Completion Rate versus Contribution Rate

Contribution rate was a significant predictor of completion rate: estimate = -0.425, $t(71) = -2.517$, $p<0.05$. A linear model predicting completion rate from contributions accounted for 7% of the variance: $F(1,71) = 6.33$, adjusted $R^2 = 0.07$ (See Figure 13.)

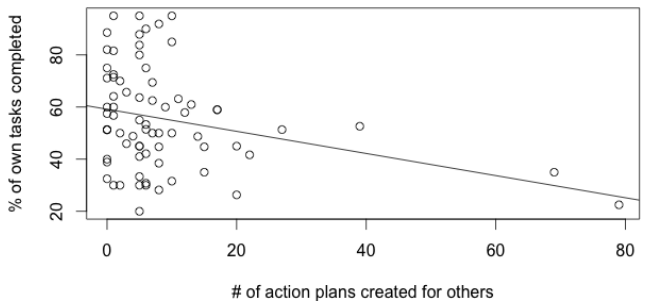

Figure 13 The more Community participants created action plans for others, the fewer tasks they completed themselves.

3.6 Study 3: Recycling Action Plans

Study 1 and 2 found that providing action plans customized to each person's tasks helps them complete more of their tasks. However, many tasks can be similar across people. Would action plans created for one person be helpful to others? To answer this question, study 3 used the GeNiLP algorithm.

Method

As a corpus, the study used 6000 tasks that were previously decomposed to action plans by the crowd. These tasks came from the participants of the Crowd and Community conditions of study 1 and 2 and other actual users of TaskGenies. The

conducted experiment was similar to study 1 and had two conditions: Recycle and Control.

Participants

Thirty-nine people, all U.S. residents and Internet users, participated in this study (15 people 18-25 years old, 16 people 26-35 years old, 5 people 36-45 years old, 2 people 46-55 years old, 1 person did not disclose their age; 13 female, 26 male). Similar to study 1, participants were recruited through an online solicitation with an iPad raffle as the incentive.

Procedure

As in study 1, participants provided 10 lingering and 10 non-lingering tasks they were planning to complete in the near future. Participants across all conditions were unaware of the TaskGenies Web site and received action plans by email.

Control: Identical to study 1. Twenty participants were randomly assigned to this condition.

Recycle: Nineteen participants were randomly assigned to this condition. GeNiLP matched each of their tasks with the most similar task from the corpus. The system reused the matched task's plan, emailing it to participants as a plan for their original task. Participants were not informed of the reuse: they were presented as personalized plans. When no tasks in the corpus had a similarity coefficient higher than 0.3, the study crowdsourced an action plan for that task. We chose 0.3 as the threshold empirically with the goal to force-match as many tasks as possible and leave

out tasks that clearly did not have a counterpart. With this threshold, 95% of the tasks were given a recycled action plan and 5% were given a custom made action plan from the crowd.

To make the Recycle condition parallel to the Crowd condition of study 1, study 3 also sent the action plans by email and each email listed one task. To ensure that the number and timing of emails were consistent with the crowd condition, each participant was randomly paired with a participant in the Crowd condition of Study 1. Measured relative to the beginning of their respective studies, when a Study 1 participant received an email, TaskGenies emailed their counterpart in the Recycle condition an action plan. The wording of these emails was the same as in Crowd condition. See Appendix A for a sample email. The rest of the method was the same as in study 1.

Dependent Measures

Like in study 1, this study measured how many tasks participants completed in a week. One week after the study began, the system sent all participants an email with a link to a Web page listing their 20 tasks. Participants were instructed to mark the tasks they completed. No other dependent measures were collected in this study.

Results

Completion Rate

Participants in the Recycle condition completed more tasks (56.2%) than participants in the Control (43.1%) condition. A pairwise t-test found a significant

effect between Recycle and Control ($t=2.21$, $p<0.05$). These numbers are based on the 95% of tasks for which the NLP algorithm created action plans, we excluded the 5% of tasks for which the crowd created new action plans.

Qualitative Analysis.

Both the strengths and weaknesses of GeNiLP (see Appendix D) come from its synonym-engine, WordNet [30]. In cases where a word in one task has a semantically related word in another task, the algorithm does well: Tasks like “prepare dinner” and “make lunch” get matched. But this approach is susceptible to a few problems. First, GeNiLP only handles single-word synonym detection. It cannot correspond a multi-word phrase, like “Call” and “catch up with”. Second, its detection is verb-centric and corresponding verbs can produce poor matches. For example, GeNiLP reports a strong match between “call the people who will fix my car tomorrow” and “fix the water heater” because they both contain the verb fix, which probably isn’t desired behavior.

3.7 Study 4: Comparing Genies with Other Approaches

A study compared Genies to three other workflows by creating action plans for the same 10 tasks. The study asked 10 people to give us one task each. Participants were recruited through an email solicitation in our university. Action plans were created for each of these tasks with all workflows. Tasks were posted to Amazon Mechanical Turk, and three workers generated plans for each task. This resulted to 30

total plans for each workflow. There was no worker overlap between workflows, and no worker could create two plans for the same task, but workers could create plans for more than one task in the same workflow. Participation was restricted to workers with 80% HIT acceptance rate and each HIT paid $0.04. A rater blind to condition rated all plans on an 11-point scale of perceived quality. The four workflows were:

1. Parallel: Workers independently create action plans.

2. Serial: The action plan of each worker is iteratively given to the next worker for improvement.

3. Revising: A worker creates a complete action plan, then rates all prior plans for the same task, then is asked to revise his work.

4. Genies: A worker rates examples of plans for other tasks, then creates their own, with the opportunity to integrate elements of other solutions.

Results

The overall rating for Genies action plans was higher than other conditions (see Figure 14). An analysis of variances was performed with workflows and tasks as factors and average action plan rating given by the blind rater as the dependent variable, finding a significant effect of condition ($F(3,36)=4.30$, $p<.05$). Follow-up t-tests with false discovery rate (FDR) correction found a significant difference between Genies and Parallel ($p<.01$), between Genies and Serial ($p<.05$), and between Genies and Revising ($p<.0001$). No difference was found between the other conditions.

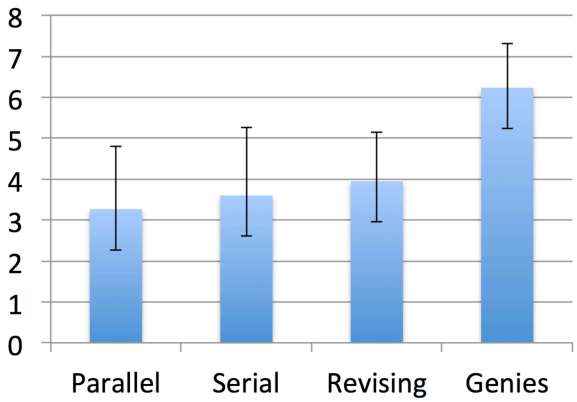

Figure 14 Average action plan quality ratings for each workflow, on an 11-point Likert scale (0 – lowest score, 10 – highest score). Error bars indicate 95% CI.

Upfront Ratings Reduced Bad Work.

The Parallel, Serial and Revising workflows suffered a lot from spam and solutions that were solving the wrong problem (about one third). For instance, some workers tried to break down the task titles into word sets. For the task "read the rest of the Hidden Reality," one worker simply tokenized it as "read" "the" "rest" "of" "the" "hidden" "reality". Another tried to execute the task, responding, "I am reading the rest of the reality which is actually hidden and mystical for sure". We revised the instructions several times but it did not eliminate this problem.

By contrast, only a couple of Genies responses contained such problems. Often, when pressed for time, people skip instructions and go directly to the task. They only return to read instructions to the extent they get stuck or confused. With Genies, unlike the other workflows, the initial rating task forced workers to understand the endeavor. These results provide an example of how interactive testing improves attention and learning [46]. Rating examples also helps set expectations and implicitly communicates that work will be peer-reviewed. Workers may believe this peer-review

affects their payment and therefore perform higher quality work. Benefits of presenting prior solutions in the middle of work

In the Parallel condition, people could not see other solutions; our review of the results suggested they suffered from it. By contrast, in Serial, people sometimes suffered from conformity: for a task, “Read the rest of The Hidden Reality”, the first worker wrote, “Week 1:Read chapters 1-3”, “Week 2:Read chapters 4-5”, etc. The following workers created refined plans such as “Read Chapter 1”, “Read Chapter 2”, “Read Chapter 3”, and so on.

The Revising and Genies workflows performed much better in that respect. In both cases, workers were first asked to create a plan from scratch, and then shown others’ plans later on. (Revising showed alternatives after participants had a complete solution; Genies showed alternatives mid-stream.) Genies mid-stream examples yielded higher quality.

Less than a third of workers actually revised their work in the Revising workflow, whereas in Genies the vast majority of workers incorporated at least one step from prior work into their solution. The activation energy needed to influence a completed solution (as in Revising) seems to be higher than the energy needed to influence a solution that is still under development (as in Genies). Figure 15 shows an example of how Genies quickly converges to good results by bubbling up good ideas.

The above results are consistent with prior research showing that examples sometimes increase the quality of people’s work [58,65,97], sometimes constrain it [84], and sometimes are surprisingly underused [34]. Presenting examples at the

beginning (as in Serial) inhibited independent thinking; when presenting them at the very end (as in Revising) or not at all (as in Parallel), it meant that people did not (or could not) leverage others' insights. Presenting examples in the middle (as in Genies) seems to offer a sweet spot: late enough that people have thought independently, yet early enough that it can still have influence [54].

Creating Plan For: "Read the rest of The Hidden Reality"

Worker 1
1.a Count the pages you have left to read in this book on theoretical physics.
1.b Set a date by which you promise yourself you will finish the book.
1.c Divide the days left by the pages remaining. That is how many pages you must read a day.
1.d Put the book on the kitchen counter or somewhaere you must walk past the book all the time.
1.e Stick to the schedule and you are done.

Worker 2
2.a (=**1.c**) Divide the days left by the pages remaining. That is how many pages you must read a day.
2.b Set aside a time every day in which you will read.
2.c Add a calendar reminder so that you remember.
2.d Go to a quiet space to read where there are no distractions.

Worker 3
3.a (=**1.b**) Set a date by which you promise yourself you will finish the book.
3.b (=**1.c**) Divide the days left by the pages remaining. That is how many pages you must read a day.
3.c (=**2.b**) Set aside a time every day in which you will read.
3.d (=**2.d**) Go to a quiet space to read where there are no distractions.
3.e (=**1.e**) Stick to the schedule and you are done.

Figure 15 Good ideas bubble up quickly using the Genies pattern. The second worker created 3 new steps and adopted 1. The third worker ended up discarding all his/her original ideas and adopted the best ideas from worker 1 and worker 2.

3.8 Discussion: When And Why Is Providing Action Plans Helpful?

Overall, providing action plans helped participants complete significantly more tasks than receiving no plans or being prompted to create their own plans, even if provided plans were reused across participants. Provision proved especially beneficial

for tasks that were considered high-level or lingering. Combining Community creation and algorithmic reuse offers the potential of a self-sustaining system without scalability concerns.

This section explores possible explanations for these results, comments on privacy concerns of this crowdsourced approach, and discusses limitations of these studies.

Effectiveness Hypothesis Revisited

Our studies found that productivity increases when high-level tasks are broken down into more concrete steps. This result supports the actionability paradigm: People work more effectively when there is a clear path forward [2,5].

We see three reasons why crowd-created plans improved completion more for lingering tasks than non-lingering ones. First, people's memory of insights, intentions, and necessary related information fades over time, which impedes delayed tasks. Plans offer concrete steps that help people recall or replace forgotten knowledge. Second, when people review a bloated task list, they practice viewing and ignoring lingering tasks. When others provide action plans for such tasks, it might break the cycle of habituated inaction. Third, having an action plan at hand lowers people's activation energy to start work.

It should also be pointed out that, sometimes, deferring tasks can be a savvy choice to focus on more important or urgent work [10]. As one participant wrote in the TaskGenies forum: "People don't have tasks lingering in their to-do lists because they don't know the steps to do them. They have things lingering on their to-do lists

because other things come up that are more urgent, continually pushing these things to the back." TaskGenies doesn't require people to work on tasks with action plans, it simply reduces the activation energy for tasks people choose to work on.

All three TaskGenies studies used a very low-fidelity version of the providing action plans intervention: they asked participants to submit 20 tasks, then sent plans to participants by email, and then asked participants to visit a page and declare which tasks they completed. A far more useful way to perform this intervention would be for it to happen within the task list of choice of the users, i.e., the place where they are already managing their tasks. Despite our low-fidelity method, people completed statistically significantly more tasks when they received plans. If commercial systems start providing action plans integrated into the experience of their users, we expect the benefits of providing action plans to have a much greater effect compared to what we saw in these studies.

Missing Context and Vague Task Titles

In our experiments, action plans provided greater benefit when the plan writer clearly understood the task. One forum participant wrote, "It's really hard to do a task decomposition for something that says 'Plan SOP.' What's that?! Make sure you tell people when they're submitting tasks to be really specific. Otherwise our recommendations will be too vague and a waste of time."

To expand the range of help that the crowd can provide, systems must capture, discover or infer some missing context. For this reason, TaskGenies allows users to

enter a reason when they reject an action plan. This reason is subsequently added to the task description for new workers to do a better job.

Aside from contextual knowledge, worker-user cultural and educational background alignment is also important. For example, for the task "declare minor [degree]", a worker created the plan "find a kid" and "declare it minor". In this particular case, the Genies pattern was able to rule out this wrong plan through rating, but one could imagine other situations where tasks require knowledge of a specific jargon to be understood.

Reusability Hypothesis Revisited

The third experiment found that reusing action plans significantly increased task completion rates. Many successful social computing systems have the property that all users benefit when a small portion does most of the work, e.g., Wikipedia [48]. TaskGenies accomplishes this through automatically recycling plans for similar tasks. Algorithmically reusing example plans also enables systems to respond immediately. Improved matching algorithms may further increase productivity.

What is the Strength of Crowd-Created Action Plans?

Looking at the plans, it seems that at least some of the time, the recipient had not thought of the plan provider's insights. CrowdPlan reported similar findings [57]. As one TaskGenies participant wrote on the forum, "For me, the best ones have been those that told me something new. Like someone introduced me to an organization nearby that meets weekly to practice speaking German (my task was to practice more). That was so helpful!"

Prior research has shown the importance of independent decision-making for achieving the wisdom of the crowd [86]. This research shows that combining independent thinking with cross-pollination can increase quality for creative tasks. Viewing examples can increase the quality of people's work [58,65], even if it sometimes reduces independence or increases groupthink [84]. As one participant wrote, "For me, breaking down the tasks into logical steps and seeing the work others had done on the same tasks was useful." CrowdPlan displayed examples in the beginning to tell the crowd worker what not to create, to avoid duplicates [57]. Our intuition in designing TaskGenies was that showing people examples in the middle of their work would provide inspirational benefits and more diverse ideas, with minimal constraining effects. Kulkarni et al.'s results [54] support this intuition.

Community Approach: Peers that Helps Each Other

Community-created action plans helped people complete significantly more tasks and reduced the workload of the crowd, but it did not fully eliminate it. Every Community participant received 20 action plans and was encouraged to create the same amount of plans for their peers. In response, some altruistic participants went on to create up to 79 action plans for others; others created a few or none. On average, participants created fewer plans than the amount of plans they received. Reciprocity and altruism were not enough to create a one-to-one ratio of production and consumption of action plans. To improve contribution, we briefly experimented with a leaderboard counting plan creations. Anecdotally, this motivated the top contributors. Exploring this remains an avenue for future work. Figure 13 indicates that the more

people contributed the less tasks they completed themselves. It is possible that participants with a tendency to procrastinate self-selected to be the ones who contributed the most.

The Genies Pattern: Benefits and Limitations

Genies trains workers on the fly by having them rate prior workers' tasks before contributing their own, and showing them plans similar to the one they're creating while they're creating it. Both potentially help demonstrate norms about the form of good action plans, such as the number of steps, instruction style and overall length.

In the Crowd and Community experiments, the Genies pattern served as a quality control mechanism that trained workers, encouraged first the divergence of ideas and then their convergence, and finally helped to select the best plans.

Future work should characterize the efficacy of the Genies pattern in more domains. How large can tasks be? Does Genies provide benefits for tasks with a correct solution like transcription or are its benefits primarily for more open-ended tasks? Potential domains for exploration include brainstorming, ideation, advisory services, copywriting, editing, poetry, image tagging, emotion and feelings capture, and theorem proving.

Automatic Reuse Lessens Privacy Concerns

Not every task can be shared online. Sharing some may be unethical, embarrassing, and in some cases possibly even illegal. Limits and safety checks are

needed in any automated, crowdsourced system. For example, the crowdsourced scanning system Cardmunch (http://www.cardmunch.com) prohibits users from scanning their credit cards.

As an empirical matter, no one has yet publicly posted privacy-sensitive tasks to TaskGenies. To reduce accidental publishing, the system requires users to opt-in: by default all tasks are private. Automatic reuse further minimizes privacy risks. For tasks in the database, users can receive an action plan without sharing the task with another person. For novel tasks, users could elect to share them and benefit from TaskGenies' plan provision system, or they could use the TaskGenies interface to create plans for themselves. If they create their own plan in TaskGenies, the NLP algorithm will select and provide example steps in the synthesize phase.

Chapter 4

EmailValet

The TaskGenies studies found that the crowd is more effective when it has enough context about people's information, but people might feel uncomfortable sharing their personal information if crowd workers have unrestricted access to it. To address these concerns, this dissertation introduces the *valet* privacy and accountability technique for crowd-powered systems[4]. It applies valet crowdsourcing on email task management: tasks are an implicit part of every inbox, but the overwhelming volume of incoming email can bury important requests. It presents EmailValet, an email client that recruits remote assistants from an expert crowdsourcing marketplace. By annotating each email with its implicit tasks, EmailValet's assistants create a task list that is automatically populated from emails in the user's inbox. The system is an example of a valet approach to crowdsourcing,

[4] This work was also presented in [52]

which aims for parsimony and transparency in access control for the crowd. To maintain privacy, users specify rules that define a sliding-window subset of their inbox that they are willing to share with assistants. To support accountability, EmailValet displays the actions that the assistant has taken on each email. In a field experiment, participants completed twice as many of their email-based tasks when they had access to crowdsourced assistants, and they became increasingly comfortable sharing their inbox with assistants over time.

4.1 Formative Survey and Interviews:

Concerns with Crowd Assistants

A formative study combined a large-scale online survey with semi-structured interviews to establish current email privacy and security concerns. The study began with an online survey on Amazon Mechanical Turk with 585 U.S. residents (59% female; 36% aged 18–25 and 33% aged 26–35). It offered a $0.01 payment, which reduced the incentive to spam and selected for workers with intrinsic interest in the subject.

The study followed up with a second round of 48 semi-structured interviews (32 male) of participants we recruited through email lists at our university and from our professional networks. Among these 48 participants, 33 were MBA students and 15 were people with technical backgrounds (five of these 15 people were at the manager level, six were students, and four were researchers). Seventeen interviews were conducted in-person; 31 were online. The interviews discussed specific concerns and opportunities with crowd-powered inbox assistants.

The study combined the interview and survey data inductively: We clustered themes from the interviews and matched them with quantitative survey results. Several authors then collaborated to code interview notes and transcripts with those themes.

Results

Respondents expressed strong concerns about sharing their inbox with a crowdsourced assistant. However, they also resonated with the goal of better task management in email and raised design opportunities around filters, whitelists, and accountability. The results from the online survey and follow-up interviews are presented below.

Email is a Popular, but Frustrating Task Management Tool

Forty-eight percent of the online survey respondents reported using emails to manage tasks. Of those, 77% sent email reminders to themselves, 47% used their inbox as a to-do list, and 41% stated that they would use an online service that helps better manage tasks in email. This aligns well with related work's findings [11]. Survey participants offered a glimpse into their day-to-day email triage: for example, "I want to be able to add little notes to every email in my inbox. I would isolate the exact action I need to take on each email." Another participant shared: "The biggest pain point is that for each email for which I need to take action, I have a specific action I want to take, but I cannot record it anywhere for the email. Every time I see the email, I go into it and re-think about what the action is, and then I decide if I have time to do the action. I do this an average of 2-3 times per email. It kills me."

Interview participants were interested in having crowdsourced assistants support inbox triage. One manager wanted an assistant to filter the inbox and "automatically archive all messages that do not require me specifically to take some action." Another participant, familiar with Priority Inbox, said: "I want an affordable […] human alternative to applying my rules to emails, i.e., what to escalate to me immediately, respond to quickly with a canned response, flag for my review quickly, flag for my response immediately, distill the action I need to take, delete, etc."

Privacy and Security Concerns

The survey responses indicate that privacy concerns were the major perceived roadblock to adopting a system like EmailValet. More than two thirds of respondents cited major concerns: they were only willing to share a few messages manually (35.4%) or share nothing at all (38.1%). Roughly one quarter (26.2%) were comfortable with a solution that determines sharing through email filters and automated rules; only a few (4.1%) were ready to share their entire inbox.

Interview participants who said they were only willing to hand-pick emails that an assistant could see acknowledged that this manual curation might neutralize any time savings the assistant might offer: "Unfortunately, the only way I'd feel totally comfortable is if I could pre-screen messages beforehand, but even the act of forwarding certain messages to an assistant […] feels too time-consuming."

Interview respondents who were willing to share parts of their inbox wanted strict access restrictions. They viewed historic inbox data as unnecessary: "[Historical access] would make me nervous about finding something I didn't remember was in

there." Most participants also wanted the inbox automatically filtered to remove personal emails and financial information and passwords; and the ability to revoke permission when a sensitive email slips through.

The more freedom the assistant has to take actions, such as write replies or archive emails, the more participants felt the need for monitoring. A manager said: "I do not mind sharing, as long as I can verify exactly how my information is used." This phenomenon has been noted previously with shared inboxes [51].

Trust was the critical concern: to share their inbox, interview respondents needed to feel they could rely on the assistants. Some suggested that they would want to "talk to them personally", get to "know them somewhat well" or "vet them myself". Furthermore, respondents wanted confidentiality assurance: "I'd want some liability insurance for what a rogue assistant might do with my information."

In summary, despite improvements in email clients, information workers still spend significant effort managing tasks in email. Respondents desired human assistance, but listed trust and privacy as core concerns, requiring mechanisms to limit potential damage. Designing for privacy concerns is important. At the same time, people's real-world practices often differ from their pre-usage estimates. Consequently, it is important to observe actual usage and not rely exclusively on survey data.

4.2 The EmailValet System

EmailValet (Figure 2) is a task-based email client that enables collaboration with crowdsourced assistants. The current prototype shares a sliding-window history

of a user's inbox with assistants who annotate messages with tasks. In principle, this approach could support other delegated tasks such as summarizing messages, negotiating meeting times, or drafting/sending replies. EmailValet makes all assistant actions visible to the user. Icons clearly identify when messages have been processed by assistants or contain tasks (Figure 16a). Actions performed on each email are displayed along with the email headers (Figure 16c). A complete log is also available to establish peace of mind (Figure 17). Our prototype is integrated with Gmail and is available at https://www.gmailvalet.com.

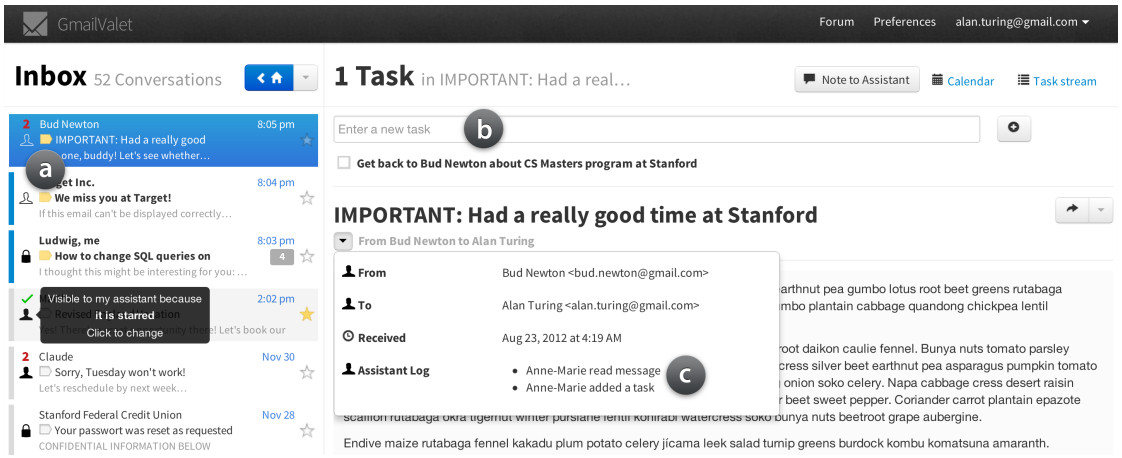

Figure 16. a) Icons to the left of the message summary indicate whether an assistant can or has viewed the message, as well as how many open tasks remain. Hovering reveals the system's reason for the current visibility setting. Clicking opens the email on the right and displays its associated tasks. Users can also add a task (b) and view actions the assistant has taken with this message (c).

In the EmailValet inbox, the left column displays email threads (Figure 16). On the right, the system presents a stream of all tasks that the user or assistant created from messages. To view an email, a user either clicks on the thread (in the inbox) or the task (in the task stream). When viewing or composing email, the message consumes the right pane; its related tasks are shown at the top (Figure 16b).

Assistants log in to a limited, 'valet' view of each inbox that they support. They can read emails and create tasks associated with those emails.

Creating and Viewing Tasks

To lower the friction for task creation, assistants and users can create a task by entering its title at the top of the email (Figure 16b). In the other direction, clicking a task shows its originating email. Icons in the inbox indicate whether an assistant can or has viewed the message, and also whether its tasks are completed (Figure 16a).

Action	Task	Message	Assistant	Time
Read message	—	[#VWJ-824-43394]: Explanation from Stanford U.	Ludwig	6 min ago
Created task	Set up unconference meeting with Martha	MyGSB Digest for the week of from MyGSB Digest	Ludwig	6 min ago
Read message	—	MyGSB Digest for the week of from MyGSB Digest	Ludwig	16 min ago

Figure 17. The log supports accountability by showing all of the assistant's activities to the user.

The right-hand task stream (Figure 2) gives users an at-a-glance overview of their tasks. Our vision is that users can treat this task stream as an action-oriented view of their email, facilitating efficient message handling. By default, tasks are ordered by recency; users can reorder them by priority.

A calendar view provides users with an overview of when tasks are due. If possible, due dates are automatically extracted using Natural Language Processing (NLP); they can also be manually edited.

Accountability and Access Control

Survey respondents were concerned about limiting assistants' access and monitoring their actions. Consequently, EmailValet introduces facilities for accountability and access control.

Accountability

EmailValet offers three monitoring techniques. First, inbox icons show whether the assistant has processed an email and extracted a task (Figure 16a). Second, the detailed message view lists all logged activities for that email: for example, opening an email, sending a response, or creating a task (Figure 16c). Third, a user can view a complete chronological log of all an assistant's actions (Figure 17). Logging doesn't prevent abuse, but does leave "fingerprints" [98]. We anticipate this log's primary benefit to be deterrence and peace of mind — like a security camera — rather than for frequent monitoring by users.

Access Control

Assistants can only view a user's mail through EmailValet. EmailValet, by default, restricts the assistant to a sliding window of the most recent 100 messages and search is disabled. Users can provide whitelist and blacklist filtering for finer-grained control. Whitelist filters restrict the assistant's view to particular labels or folders, such as starred messages, Gmail's Priority Inbox, or messages sent to oneself. Furthermore, users can provide blacklist filters to exclude sensitive messages, such as emails from family, passwords, or financial information (Figure 18).

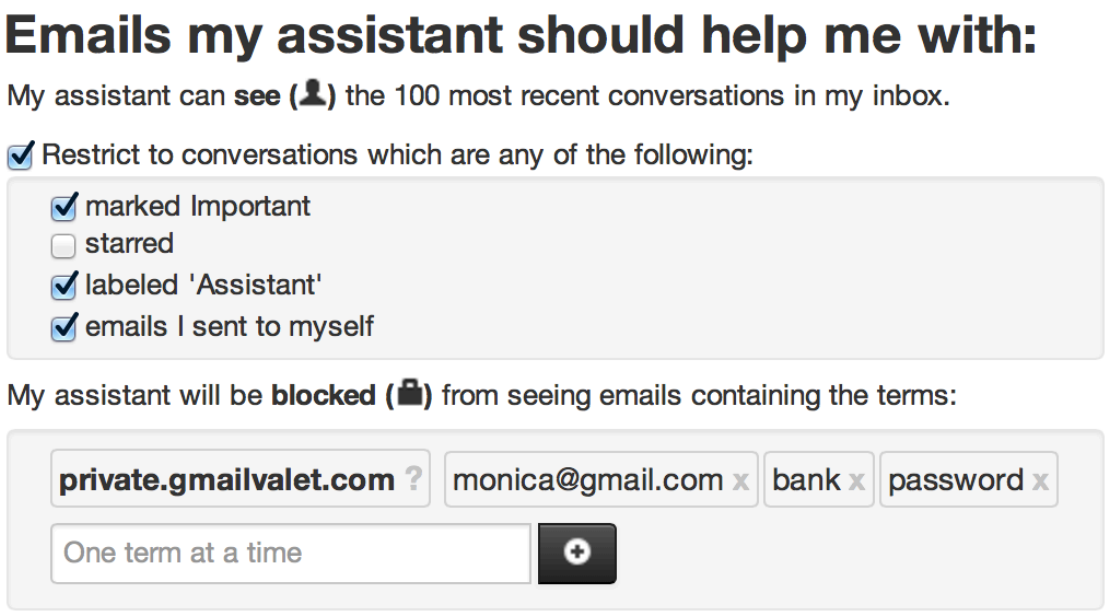

Figure 18. For privacy, users can specify rules for which emails will be visible to their assistant.

These restrictions attempt to balance the assistants' need to understand contextual connections with the user's desire not to expose their whole history to the crowd. Assistants' actions are also limited; EmailValet's current policy prevents assistants from deleting messages.

Finally, EmailValet also integrates with automatic approaches. To illustrate this, with the EmailValet prototype, users can restrict assistant-visible emails to Gmail's Priority Inbox.

Feedback and Learning

The tasks that assistants create may not always be the tasks that users want. Tasks mean different things to different users: this is one reason automated approaches often fail [32]. To provide assistants feedback, the simplest way for a user to remove a task is to decline it (Figure 2). EmailValet shows assistants which tasks were accepted

or declined, and encourages users to add an explanatory comment to the assistant (Figure 19). To help assistants learn what they missed, they can see tasks that users create themselves.

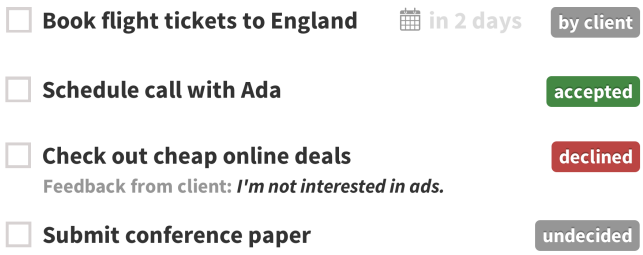

Figure 19. The assistant's view of the task stream. Feedback helps the assistant to learn the user's intentions: accepted and rejected tasks, freeform text, and user-created tasks.

To frame discussions, users can leave an introductory note about their preferences. This note is especially helpful for providing new assistants with context and user-specific heuristics. For example, users may want to emphasize certain senders or ask the assistant to apply labels to their tasks (e.g., "put [Event] in front of every event").

Assistants and users can also open a chat window to clarify any confusion. While a few crowdsourcing systems provide limited interactive feedback with requesters [26], the author is unaware of other crowd-powered systems that support interactive worker conversation with end users. User trials found that chat helps efficiently achieve common ground:

Assistant:	Do you want me to create tasks from [name]?
User:	Yes, please.
Assistant:	Ok I will, I made one and it got declined so I just wanted to make sure.

Accepted and declined tasks, notes, and chat transcripts also serve as a simple way for institutionalizing the experience and knowledge that assistants gain about each user over time. When a new assistant is called to start serving a user, they can spend some time reading this data to familiarize his- or herself with that user.

Multiple Users per Assistant and Vice Versa

Assistants can help many users simultaneously, increasing affordability for users and labor opportunities for assistants. To provide easy access to multiple accounts, assistants can switch between users with a drop-down menu. Awareness indicators on this menu show users with unread messages.

To increase system reliability, multiple assistants can be assigned to one user. This way when one assistant is unavailable or offline, another can continue processing emails. To avoid replication of work, the same tasks are visible to all assistants, and when one assistant reads an email, this email is marked ‘read’ for all assistants.

4.3 Study Comparing Email, Self-, & Auto-extraction

A weeklong field study investigated whether EmailValet helps users complete tasks and whether its privacy and accountability features satisfy users’ concerns. The study found that EmailValet accurately extracted tasks from email, that users found value in the system for their task management, and that users became increasingly comfortable with EmailValet’s privacy tradeoffs.

Method

A one-week long field study deployed EmailValet, recruiting twenty-eight participants using mailing lists at our university. Six were MBA students and twenty-two were university students of other majors, mostly technical. The study offered participants a $50 gift certificate. The study hired three online assistants through the oDesk crowdsourcing marketplace: one from Illinois, two from California. Two were work-at-home mothers. Assistants were compensated at $8 per hour to process all shared emails during the study. To control for mistakes due to large cultural differences, participants and assistants were all residents of the United States.

Participants began by authoring whitelists and blacklists that determined the assistants' access control. Each participant was instructed to use EmailValet at least twice each day to process their new emails and tasks. At the conclusion of the study, participants completed a survey focused on EmailValet's qualitative usefulness, privacy, and assistant quality.

A within-subjects experiment investigated whether EmailValet helps users complete the tasks in their email. Participants rotated through three interface conditions. Following a one-day warm-up, each participant spent two consecutive business days in each condition:

Control: Participants could not see assistant-created tasks or create their own tasks.

Self: Participants could not see assistant-created tasks. However, they could create their own tasks.

Assistance: Participants could see assistant-created tasks and create their own tasks. Participants could give task feedback to their assistant.

The order of conditions was randomized for each participant, in a Latin square manner. Participants could always read and write email. The assistant extracted tasks from emails in all three conditions. However, in the Control and Self conditions, the assistant did not receive feedback on tasks from the participant because the participant could not see their extracted tasks. At the end of each condition, participants saw any previously-hidden tasks that the assistant created. To produce ground truth, participants then accepted or rejected these tasks, gave feedback, created any tasks the assistant missed, and marked whether or not they had completed each task.

The experiment measured the percentage of new tasks that the user marked completed while each condition was in effect. This metric combines the tasks that the assistant authored and the user accepted with the tasks that the user authored themselves. If the assistant's tasks were hidden during the condition (e.g., Self or Control), the study counted tasks that were accepted during the end-of-condition review. We manually merged any duplicate tasks that the user and assistant both created. Users and assistants may make extraction mistakes, so this metric may not represent every task in the inbox.

The study hypothesized that users would complete more email-based tasks with EmailValet's task extraction than when they must extract tasks on their own (Assistance > Self) or cannot extract any tasks (Assistance > Control). The study

further hypothesized that the discipline of self-managed task management would still produce some benefit: Self > Control.

Results

Of the 28 participants who filled out the final survey, 16 consistently and successfully participated in all three conditions and thus had measurable task completion rates. This section analyzes qualitatively and quantitatively: 1) the assistants' accuracy at extracting tasks, 2) users' behavior and feedback regarding EmailValet as a task support tool, and 3) users' feedback to the privacy concerns and functionality in EmailValet. Figure 20 displays examples of assistant-extracted tasks.

- *Schedule to meet first week of September with Priya*
- *Complete flyer by the end of the week*
- *Print out flight information*
- *Send International Student Advisor a scan of my I-20*
- *Review Emily's ideas and if possible add onto it*
- *Lunch with Al*

Figure 20. Tasks extracted by assistants during the study.

Assistants' Accuracy

Measuring the precision and recall of assistant-created tasks offers a better understanding of how accurate the assistants were at extracting important tasks for users.

Precision measures the percentage of assistant-created tasks that participants accepted. On average, users accepted 71.6% of the tasks extracted by assistants. This ratio was very similar across all three assistants ($\sigma = 3.5\%$). Precision rose throughout the study, starting at 62.1% on the first day and ending with 84.8% on the last day. The rise in precision is most likely due to participants' feedback regarding tasks and to

the assistants learning more about their users. One assistant noted, "it has become easier to extract good and accurate tasks from my clients' emails over time. I feel I have gotten to known [sic] my clients better and understand the conversations better".

The final survey asked a free-response question: were the assistants' tasks relevant, or just busywork? About two thirds of participants (19 of 28) found assistant-created tasks to be of value and worth completing. Notably, four of these participants praised the assistants' creations as being on the same level as their own; one thought the assistant was even better than themselves ("they used more detail than I did") and three lauded the assistants for extracting more tasks and events than they would have done ("made the week easier"). However, some participants felt that the assistants were overeager, complaining that assistants created tasks from irrelevant mailing lists, created tasks that were too ambiguous or missed tasks completely. On the other hand, a participant explicitly called their assistant "too conservative" and would have preferred to receive more tasks. Users typically did not mind false positives: as a participant wrote, "Deleting [tasks] was easier than creating."

Recall measures how many tasks the assistant missed: the number of accepted tasks that the assistant authored divided by the number of accepted tasks that either the user or the assistant authored. This measure only includes tasks created during the Assistant condition because this is the only time that users could see the assistants' tasks in real time and add any missing tasks. By this calculation, recall was 68.6%. Often, the user logged in before the assistant had time to process the new email, so eventual recall may be higher.

The study asked users a free-response question: were they confident that their assistants would not miss important tasks? We coded their responses and found that more than half of participants (17 of 28) felt they could fully or almost fully rely on their assistant, five participants had mixed feelings, and six had negative impressions. Participants praised assistants' consistency ("after the first few days they established that they could keep track of impt [sic] details"). Missing contextual knowledge was the most common cause of missed tasks: "Many important tasks (that are not obvious) are not extracted."

EmailValet's Usefulness

Users found the assistants to be generally accurate, but did the system help those users manage their tasks? The study compared the percentage of tasks completed in each condition and supplemented the results with qualitative feedback.

Of the sixteen participants who consistently used EmailValet through all study conditions, users completed a significantly higher percentage of tasks in the Assistance condition (M = 58.4%, SD = 34.4%) than in the Self condition (M = 29.3%, SD = 28.7%) or Control condition (M = 26.3%, SD = 34.5%). A repeated-measures ANOVA was performed with condition as independent variable and task completion rate as the dependent variable, finding a significant effect of condition, $F(2,30) = 4.483$, $p<.05$. Posthoc pairwise t-tests with false discovery rate (FDR) correction found a significant difference between Assistance and Control, $t(15) = 2.89$, $p <.05$, and between Assistance and Self, $t(15) = 2.49$, $p <.05$. No difference was

found between Self and Control $t(15) = 0.24$, n.s. So, participants completed more tasks in the Assistance condition than in the Self or Control condition.

Participants were enthusiastic about the positive impact EmailValet had on their task management. When asked if they would like to continue using EmailValet, one participant replied, "any help in making sure everything gets done would be greatly appreciated." Another insisted that they would like to use it only with the assistant: "What I need is an extra pair of eyes." A third participant praised the assistant's work because their tasks were "like magic": "very convenient and much easier than doing it myself." For users who didn't feel that assistants supported their task management, they still found that the task-centric interface kept them aware of incomplete tasks. Roughly 40% of the participants wanted to use EmailValet as their main task list following the study (12 of 29), and several continued using it voluntarily the week after the study concluded. Those who were uninterested in continuing felt that it needed deeper integration with other PIM tools for them to fully integrate it into their workflow.

Privacy Concerns and Trust

The study also asked participants to report which emails they made available to assistants and which they blocked. Only two participants opted to manually whitelist individual emails for the assistant to see. Most participants shared everything or whitelisted a set of filters (e.g., priority messages, starred messages, school, work). Those who filtered for privacy blacklisted patterns such as passwords, banking information and more intimate contacts. As one participant put it: "I thought the only

way for the service to be most helpful would be to ensure they could look at everything." Some participants loosened their settings over time: as one put it, "Originally I did not share emails from my boyfriend [name], but I changed that after realizing he sent me emails with tasks for me to do".

A majority of participants (18 of 28) initially felt somewhat uncomfortable entrusting an assistant with their emails. However, over half of those with concerns (10 of 18) and over a third of all participants (11 of 28) reported that they felt more comfortable with the service over time, while no one reported a decrease in comfort. Reasons for the increased trust were diverse: One participant named the first task extractions ("I felt more optimistic and was pleasantly surprised […] it was surprisingly useful and effective"). Another one found the assistant biography helpful ("my assistant was a mother"). A third simply got accustomed to what she called a "slight breech [sic] of privacy": "In the beginning, it felt weird that my assistant was reading personal emails from my family and then creating a task for me to do based on it, but I didn't really feel strongly enough about it to change my privacy settings... I just kind of went with it."

Assistants shared the discomfort, as they were initially not at ease with going through their users' emails and occasionally stumbling upon sensitive information: "Some emails I felt like I was invading their privacy because it was emails between my clients and their family, but I didn't react anyway because it's a job I have to do". Similar to users, assistants became more comfortable over time.

Despite this increased level of comfort, almost two thirds of participants (19 of 28) said the assistant did not care for them personally but was simply "doing their job," and only one participant felt the assistant personally cared. This was not necessarily bad: four participants explicitly stated that they did not want to get to know their assistant. However, seven other participants suggested more communication with the assistant so that they could feel more like the assistant personally cared. Similarly, opinions were split on whether knowing the assistant made it more or less comfortable share your inbox.

Assistant Economics

The assistants recorded 70 hours of work during the study. They processed 12,321 messages — 8,679 for the 28 participants and the rest for 10 existing users of the system who did not take part in the study — and created 779 tasks. On average, each assistant processed 2.94 messages per minute. This rate would extrapolate to 1,408 messages in an eight-hour workday if the assistant were fully occupied. On average, assistants created 6.32 tasks per 100 emails they read. Participants received 38.8 messages per day on average, fewer than recent studies have reported [31].

At this rate, one assistant could support 36 users simultaneously in an eight-hour workday. Such support would cost each user $1.78 per day. However, in our deployment, each assistant was assigned 12 users on average. We asked the assistants to tell us how many users they felt they could support. They generally felt that they had additional bandwidth. For example, one said, "[I have] 10 current clients, it's pretty easy to keep track of the clients, […] so ideal would be probably 20 to keep me

busy more". The author of this dissertation used the system for over six months and found himself asking his assistant to perform more and more actions on his email over time (including archiving or starring messages, scheduling meetings, and executing some easy tasks), spending an average of about 40 minutes per day. At this rate, an assistant can effectively support a maximum of 12 users.

Limitations

This study has several limitations. First, the participants are largely young and technology-literate. Second, participants self-selected to participate in the study. Third, the study's short duration captures the first days of usage, but usage patterns may evolve. A longitudinal evaluation with a broad population is a clear next step.

4.4 Discussion

EmailValet extends crowdsourcing to tasks that require access to private or sensitive data. This section reflects on the implications of this decision, the importance of context and common ground, and on a future vision of massively multiplexed crowd assistants.

Adaptation to the Privacy Breach

Fully 70% of the formative survey respondents reported they would be unwilling to share significant fractions of their inbox. Yet, after a week of use, only 10% of participants cited privacy as a reason that someone might abandon EmailValet. Even considering sampling differences, this is a major shift. Furthermore, participants rarely used EmailValet's accountability features: the features were more useful as a

security blanket than for functionality. While engineers and designers often describe privacy as a property — a system either has privacy or it does not — in practice, privacy is dynamic [73]. When the benefit exceeds the invasion, we share information that we previously considered private.

Participants in the EmailValet study did not personally know the assistants. Also, participants were not family, but there may have been participants who knew each other. Even though we do not have evidence of this occurring, these participants could have exchanged private information about other participants over emails. If these participants happened to have the same assistant, then this assistant could gain some knowledge about a user that the user himself is not supposed to know. To avoid accidental information leaks, commercial versions of EmailValet have to carefully design mechanisms to avoid this type of situation. At minimum, the assistants should be trained on how to be mindful and tactful.

The Assistants' Lack of Context

Traditional crowdsourcing systems focus on context-free micro-tasks (e.g., [13]). However, supporting personal and private tasks requires rich context [27]. There were several situations in which EmailValet assistants lacked this context and produced incorrect tasks. The assistants recognized this challenge: in the future, they asked for a short fact file about each user, including their profession and place of residence. We are eager to let assistants flag messages that they find useful for understanding each user, and to allow them to write or share notes between each other. These notes institutionalize assistant knowledge, which would enable a smooth

“changing of the guard” as assistants begin and end their work. Companies can also lessen the context and privacy problems by adapting a model midway between EmailValet’s distributed experts and in-house administrative assistants.

Tasks and Busywork

Users completed more tasks in the Assistance condition, but is this because EmailValet focused them on unimportant busywork? There is some evidence that this may be happening: one participant explained that they accepted “tasks or events I didn't necessarily plan to do [...] because it was still good to have it there [as] a reminder, whereas I wouldn't have bothered to make note of that task myself.” These tasks were likely middle priority: not as unimportant as busywork, but not important enough to be the top priority. In general, however, it is up to the user to decide whether an assistant-created task is important or busywork. The framing for this decision can be communicated to the assistant, helping them adapt to each user’s needs.

From Personal to Massively Multiplexed Assistants

Online assistants may be affordable enough for most information workers to use in small amounts. Likewise, assistants can multiplex their services across multiple users. Are we seeing the beginnings of democratized access to executive assistants? Not yet: several issues remain and demand workable solutions. First, as our formative work revealed, people are reluctant to entrust online assistants with access to their sensitive data. As systems like EmailValet establish trust and produce high-quality work, these concerns may lessen. Second, the quality of an online assistant may not

compare to that of an executive assistant entirely dedicated to one person. Users of EmailValet-style systems thus face a trade-off between cost and quality. However, collective intelligence of multiple assistants may produce higher quality than any dedicated assistant. Third, as this dissertation found, missing context remains a major struggle for many assistants. The author used the system for over six months with a few different assistants. While all assistants were able to extract reasonable tasks, assistants with the same cultural and educational background were also able to execute some of the tasks and reply to some of the email messages directly on his behalf.

Finally, systems like EmailValet can easily measure the individual performance of each assistant. This could be accomplished by tracking metrics such as their task extraction speed and accuracy, the number of times they ask users for clarification, and the number of missed tasks, or even by occasionally surveying users to rate the quality of the work they receive. These systems can in turn use this information to retain the best assistants and replace or train underperformers.

Extended Usage of EmailValet

In this study, EmailValet tested only one of the many possible uses of Valet crowdsourcing in email management. One of the participants of the interviews described in Chapter 4.1, became a regular user of EmailValet and provides an example of other uses. This participant asked his assistant to archive any emails that are not sent directly to him from humans that appear to personally know him. He also asked the assistant to star messages that appeared to be urgent. Note that this

participant was excluded from participating in the EmailValet study described in section 4.3.

Another user of EmailValet decided to use the system with his own local assistant, a regular employee at the same company. In addition to sharing with the assistant a subset of his email through EmailValet, he also shared his calendar and asked the assistant to reply to as many of his messages as she possibly could, using his calendar. EmailValet allows users to decide whether an assistant can send email from the user's account or not and whether the sender will appear as the user or the assistant.

The author of this dissertation heavily used EmailValet during the writing of this manuscript to triage and prioritize his email correspondence. During this time he replied to very few emails every day using the following process. His assistant extracted tasks for him and managed the priorities on his task list. Every now and then instead of checking for new email he looked at the top of his task list, and if there was a task that seemed urgent or important he clicked on it and read the related email and responded right away without reading the rest of his inbox. The author also instructed his assistant to reply to some emails on his behalf when possible. Occasionally, the assistant consulted the author with questions such as "Should I reply to X with a similar response that I gave to Y?", which further reduced the author's workload.

Why is Having EmailValet Extract Tasks Useful?

EmailValet uses online assistants to extract tasks from peoples' inboxes. This saves people the time that it would take them to do so themselves. However, an even

greater benefit is derived from the availability of the tasks themselves. As the Self condition demonstrated and related work has found [11,63,99], people tend to inadequately extract tasks from their emails. In other words, extracting tasks to a task list is meta-work (work about work) for which people have a tendency to avoid. EmailValet provides this organization "automatically" and consistently. The consistent presence of an up-to-date task list is much more beneficial than merely saving the amount of time assistants uses to compile the list. This way, users enjoy the full benefits of reduced activation energy to begin working on tasks and confidence that no task has been forgotten in their inbox other than the ones neatly organized in their task list.

Chapter 5

Conclusions and Future Work

This dissertation introduces the Genies pattern for organizing crowd work and the Valet pattern for providing privacy and accountability. The TaskGenies and EmailValet systems instantiate these patterns in two core aspects of personal information management – email and tasks. This thesis has shown that overloaded information workers benefit from external management of their personal information. This dissertation has also shown that crowdsourcing is an effective way to provide this external management.

To conclude, this chapter will review the main contributions of the thesis and consider important avenues for future work.

5.1 Summary of Contributions

This thesis has presented design and implementation patterns for crowdsourced systems and presented behavioral findings. TaskGenies (Chapter 3) demonstrated that automatically providing action plans helps people complete more tasks. An

experiment found that action plans were particularly valuable for high-level and lingering tasks. The TaskGenies system supported these experiments and demonstrated that crowdsourcing is an effective way to create action plans. To scale plan provision, TaskGenies introduced a community approach in which users create tasks for each other. Further scaling was accomplished by combining human creation with algorithmic reuse. TaskGenies also introduced an NLP algorithm that identifies similar tasks and experimentally demonstrated its utility.

Combining community creation with automatic reuse enables crowdsourcing systems to handle both the fat head and long tail common to many information domains. It also naturally adapts to change over time. We believe that such hybrid approaches may be valuable for crowdsourcing more broadly. Future work can further explore and evaluate alternative techniques for blending crowd, community, and reuse approaches.

Chapter 4 introduces valet techniques that preserve privacy and accountability for sharing private data with crowdsourced assistants. These techniques take shape in EmailValet, an email client that recruits crowdsourced human assistants to extract tasks from each email in the inbox. As a valet interface, EmailValet focuses on: 1) parsimony, providing just enough access for assistants to extract tasks and giving users rich controls to limit assistants' permissions; and 2) visibility, surfacing all assistant actions within the interface. In a one-week evaluation, EmailValet users completed about twice as many tasks when they had access to the assistants' help.

5.2 Methods and Challenges

There are many challenges in conducting research on personal information management and its related domains involving people's everyday life. First, it is difficult to design controlled experiments: to persuasively conclude that the intervention under study is effective for the participants' actual data, the intervention needs to be tested with the actual data of each individual participant. A standard controlled experiment would ask participants to operate on a standardized inbox or task list, but using data that is not their own would not be sufficiently conclusive to extrapolate the result to their own data. On the other hand, trying to intervene on people's own task list and inbox suffers from (i) potential confounds: for instance, the confound of email notifications in Study 1 of this dissertation which we addressed by introducing the Prompt condition; and the confound of the novelty effect of using a new email system in the EmailValet study which we addressed by introducing the Self condition; (ii) reluctance of people to participate in the studies due to concerns about the privacy of their data; and (iii) the intrusive nature of intervening in people's personal information management: people are accustomed to their own tools and practices and may not want to change, even for a week or two. This last reason also makes it difficult to run long-term studies as well: even if people find the intervention of a novel system useful they may not want to use it long term for other reasons, such as aesthetics, the absence of other core functionality such as a mobile interface, habits, or the mature polish of commercial products already on the market. A lot of engineering effort went into putting EmailValet on par with existing email clients and

creating a mobile interface for it before people were willing to use it, even for a week-long study.

Building fully featured consumer systems, like to-do lists or email clients, requires substantial engineering effort. An alternate way to do research on such systems is to modify existing open source software or utilize widget APIs to modify commercial systems when available. Field and observational studies can also provide insights about how people use these systems, for example, one can ask permission from users to use remote screen sharing tools to capture their habits. Of course, these techniques have limited intervention capabilities, but they are easier to deploy. Finally, asking people to participate in lab experiments using mock-ups is another alternative.

5.3 Implications

Privileged individuals have long relied on assistants to organize their personal information and help them be more productive. Crowdsourcing has the potential to democratize personal assistants. This empowerment can help people be more productive and have more time to spend with their families, and it can benefit many people, including those who work as assistants themselves. Just as a restaurant waiter can become a restaurant client by going out to eat when their shift is over, crowd assistants can also utilize the crowdsourcing service they are a part of to have their own personal information managed as well. This is only possible with crowdsourcing and its tremendous price reduction benefits; it is unlikely that a traditional personal assistant would have the resources to pay another personal assistant to help them with their own tasks.

For crowdsourcing to be a sustainable a form of labor, cost reductions must come from the efficient use of the crowd and not from underpaying workers. Through the efficient utilization and specialization of workers, we believe that it is in fact possible to increase salaries and the reputation of workers while also reducing the cost for users.

5.4 Systems and Domains

Genies and Valet crowdsourcing can be applied in fields other than PIM. One potential application is enabling individuals to communicate with a large number of people on a one-to-one basis. Imagine the Prime Minister of a country was able to individually communicate with one million civilians, or a musician talking personally with all of their fans, or a researcher being able to conduct in-depth interviews with ten thousand informants, or a web-based company emailing all of their users from a non- "no-reply" email address. Today, the state of the art for these kinds of interactions is no-reply newsletters, public forums, online surveys, or one-way email blasts that are followed up with other one-way blasts thanking recipients for their "overwhelming response". With Genies and Valet crowdsourcing it may be possible to create systems where algorithms and crowd assistants can handle the majority of the work of classifying similar responses and personalizing messages, and users only have to do high-level communication planning, writing a few appropriate message fragments or templates, and interpreting the results of the recipient reactions.

5.5 Patterns

Crowd-powered computer systems is a relatively young field of study. In addition to the Valet and Genies crowdsourcing, many more patterns need to be invented to push these systems forward and make them even more useful. The field needs patterns to assist in the institutionalization of knowledge about individuals, such as patterns for crowd assistants who are changing shift or departing the system to hand over the learning and implicit information about a user to other crowd assistants. Also, patterns need to be established to ensure that crowd workers receive fair payment, and to ensure effective rating systems are in place to support fair payment for workers and quality work for users. Additionally, patterns need to be devised to facilitate scaling across verticals; thusfar, crowdsourcing has focused on workers with general knowledge, but new patterns will be needed to apply existing crowdsourcing methods to fields with specialized knowledge, such as medicine, law, education, design, or research itself.

5.6 Behavior

Valet sharing imposes a number of privacy and accountability safeguards by limiting worker functionality. However, it is possible that prohibiting bad behavior in this way may conversely encourage it by making it a challenge. What social or legal contracts between assistants, users and systems need to be established to further

address such concerns? A cautionary example is Airbnb, a service that allows people to rent out their homes as mini-hotels. They had operated without incident for over a year until a user's house was trashed, which shocked the service's user base[5]. Similarly, crowdsourcing of personal data can provide a paper trail of which assistant is responsible for a transgression, but we still need law, contracts, and other mechanisms to address the consequences of potential transgressions.

The more crowdsourcing gets integrated into people's lives, the more we need to study the behavioral implications of it, for both the users and the workers. Crowd-powered systems can offer services that were not possible before, such as assistance with task, email and calendar management. Can they also be used to facilitate other types of positive behavior changes, such as exercising more, living and eating healthier, being happier, becoming more productive, living longer, reducing stress, getting more satisfaction from work, being more connected to family and friends, earning higher pay, or protecting the environment by reducing commuting or by positively altering people's attitude towards it?

5.7 Future Work

Crowdsourced assistants could support many additional personal information management activities. Email assistants could create action plans for tasks [51], identify meeting requests, link references to other threads, mark priorities, and write

[5] http://techcrunch.com/2011/07/27/the-moment-of-truth-for-airbnb-as-users-home-is-utterly-trashed/

personal reminders. These extensions create an opportunity to design alternative consumption methods for the inbox, including a compact, glanceable stream of events. The assistant could also support other personal information management tools: curating the user's calendar, maintaining a contacts list, or organizing appointments.

Looking further ahead, one might have the crowd, the community, or algorithms automatically execute (parts of) the action plans. For example, "buy The Hitchhiker's Guide to the Galaxy" might be executed by an algorithm, "change car oil" might be executed by local workers, "plan my trip to CHI" might be executed by online workers, and "choose a gift for Mary's birthday" might be done by the user's peers, friends and family. By formalizing the assistants' actions, we can train machine learning systems and improve the automatic approaches to task extraction.

In this work, crowd-provided action plans worked best when they required little contextual knowledge and valet assistants became more accurate as they built better common ground with the users they supported. Looking forward, algorithms or crowd workers could elicit additional contextual information when necessary. Context-aware routing – using a social network [38], location, etc. – may also improve quality and relevance. Future work can also explore whether and how people adopt plan suggestions differently depending on the source of the suggestion: are people more influenced by plans listed as human-created or personalized? Are plans from friends more valuable? And are creators more motivated to offer plans for their social network?

Additional materials about this research are available at http://kokkalis.com

Appendices

Appendix A: Email to Crowd Condition

One of the tasks you gave us was: [Task Title]

Someone suggested that they'd follow the steps below to complete this task:

[Action Plan]

Will you follow some of the steps above to complete this task?

Can you come up with your own step-by-step plan?

Before you close this email, could you possibly start this task by completing its first applicable step right away?

Write remaining steps in your to-do list, so that you can complete the bigger task one step at a time.

Appendix B: Email to Prompt Condition

One of the tasks you gave us was: [Task Title]

Someone suggested that you spend a few minutes trying to break this task down into smaller steps.

Does coming up with steps help you complete this task?

Before you close this email, could you possibly start this task by completing its first applicable step right away?

Write remaining steps in your to-do list, so that you can complete the bigger task one step at a time.

Appendix C: Email to Community Condition

A study participant created an Action Plan for you.

Action plan for your task: [Task Title]

[Action Plan]

1. Before you close this email, could you possibly start this task by completing its first applicable step? Start with a step that takes just 2 minutes (from the steps above or your own steps)

2. Create at least 5 Action Plans for others by visiting: [URL]

(click “give help” or select the tasks you want to work on)

Appendix D: The NLP Algorithm for Task Recycling

This appendix, first summarizes the overall NLP algorithm, then sketches two essential steps of the algorithm: word-sense disambiguation and computation of the similarity coefficient between two tasks.

Overall Algorithm

1. Perform word-sense disambiguation for every task in the database (offline).

2. Perform word-sense disambiguation for the input task.

3. Compute the similarity coefficient between the input task and every task in the database.

4. Return the task with the highest similarity coefficient.

Word-sense Disambiguation for All Words of a Task

For each word w in task t, ignoring stop words: (i) Use the Stanford Dependency Parser [24] to identify all modifier words M(w) of w within t. (ii) Use WordNet [30] to find each sense sw,i of w, (iii) For each sw,i compare the definitions and examples of sw,i with the definitions and examples of all senses of the words in M(w) and count the number of agreements, (iv) Select the sense sw,i with the most agreements as the most likely sense of w.

Computing the Similarity Coefficient Between Two Tasks

Intuition

We approach the similarity computation between two tasks as the maximum-weight bipartite matching between two tasks, with the disambiguated senses as the nodes and the sense similarity as the edge weights.

Pseudocode

Phase1 : Compute the matching matrix

```
FOR EACH sense x in the first task {
FOR EACH sense y in the second task {
IF two senses are directly comparable
RETURN the similarity with respect to WordNet taxonomy
ELSE { #(e.g., noun vs verb / verb vs verb in different taxonomy)
        Find the set X of synonymous senses for x
        Find the set Y of synonymous senses for y
        RETURN the ratio of their intersection
# (i.e., |X intersect Y| / |X union Y|)
    }
}
# Phase2 : Do maximum-weight bipartite matching
FIND the maximum-weight bipartite matching
NORMALIZE the final matching weight into a uniform similarity coefficient between
0 and 1
```

Examples of Matches

Study 3 set the similarity coefficient such that the algorithm force-matched about 95% of participant tasks to a corpus of 6000 tasks. Some good and bad examples of tasks matched are presented below.

Great Matches

Study for 15-122 midterm → study for midterm

Buy groceries. → buy groceries

Buy tickets to New York → Buy flight tickets

Attain a decent sleeping schedule. → Fix sleeping schedule

Medium Matches

Design an experiment and write a paper about it. → Write research paper

Find out about tax laws for earning money abroad. → find someone to do my taxes

Meet an old friend → meet my best friend

bring a relative home from airport → confirm my transportation from the airport back home

Bad Matches

Start working on Blender for simulation → start working out

Upgrade my PC → turn on a pc

Replace speakers in the car → get new car

Practice accepting my feelings → Practice my French

No Match Found

searching about some universities

find/compose poem for Friday night get-together

do my laundary [SIC]

getting to 68 Kg

References

1. Ackerman, M. A., Malone, T. W. Answer garden: A tool for growing organizational memory. *SIGOIS 1990*, MIT, 31-39.
2. Ahmad, S., Battle, A., Zahan Malkani, Sepander D. Kamvar, The Jabberwocky Programming Environment for Structured Social Computing, In Proc. *UIST 2011*.
3. Allen, D. Getting Things Done: The Art of Stress-Free Productivity. Penguin, New York, NY, USA, 2002.
4. Allow Someone Else to Manage Your Mail and Calendar. http://office.microsoft.com/en-us/outlook-help/allow-someone-else-to-manage-your-mail-and-calendar-HA010355554.aspx.
5. Amabile, T., Kramer, S. *The Progress Principle.* Harvard Business Review Press, Boston, MA, USA, 2011.
6. Bandura, A. & McClelland, D. C., *Social learning theory*. Englewood Cliffs, N.J.: Prentice-Hall, Oxford, England 1977.
7. Barley, S.R., D. Meyerson, and S. Grodal. E-mail as a Source and Symbol of Stress. *Organization Science 22*, 4. 2011.

8. Baumeister, Roy F., et al. "Ego depletion: Is the active self a limited resource?." Journal of personality and social psychology 74 (1998): 1252-1265.
9. Bekkerman, R., A. McCallum, and G. Huang. Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora. *Technical Report IR-418*. 2004.
10. Bellotti, V., Dalal, B., Good, N. et al. What a to-do: studies of task management towards the design of a personal task list manager. *CHI 2004*, ACM Press (2004), 735-742.
11. Bellotti, V., Ducheneaut, N., Howard, M. et al. Taking email to task: the design and evaluation of a task management centered email tool. *CHI 2003*, ACM Press (2003), 345–352.
12. Bennett, P.N. and J. Carbonell. Detecting Action-Items in E-mail. *Proc. SIGIR 2005*. 2005.
13. Bernstein, M., Little, G., Miller, R.C. et al. Soylent: A Word Processor with a Crowd Inside. *UIST 2010*. ACM Press, 313–322.
14. Bernstein, M., Tan, D., Smith, G., et al. Collabio: a game for annotating people within social networks. In *Proc. UIST 2009*, ACM Press, NY, 97–100.
15. Bigham, Jeffrey P., et al. "VizWiz: nearly real-time answers to visual questions." Proceedings of the 23nd annual ACM symposium on User interface software and technology. ACM, 2010.
16. Boone, G. Concept Features in Re:Agent, an Intelligent Email Agent. *Proc. AAMAS 1998*. 1998.

17. Brandt, J., Dontcheva, M., Weskamp, M., Klemmer, S.R. Example-Centric Programming: Integrating Web Search into the Development Environment. In *Proc. CHI 2010*, 513–522.

18. Brooks Jr, Frederick P. "The computer scientist as toolsmith II." Communications of the ACM 39.3 (1996): 61-68.

19. ClearContext tames Outlook. http://news.cnet.com/8301-17939_109-9947065-2.html.

20. Corston-Oliver, S., E. Ringger, M. Gamon and R. Campbell. Task-focused summarization of email. *ACL 2004 Workshop: Text Summarization Branches Out*. 2004.

21. Crawford, A.B. Corporate Electronic Mail — A Communication-Intensive Application of Information Technology. *MIS Quarterly 6*, 3. 1982.

22. Create Tasks and To-Do Items. http://office.microsoft.com/en-us/outlook-help/create-tasks-and-to-do-items-HA001229302.aspx.

23. Dai, P., Mausam, Weld, D.S. Artificial Intelligence for Artificial Artificial Intelligence. In *AAAI 2011*.

24. de Marneffe, M.-C., MacCartney, B., Manning, C.D. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proc. LREC 2006*, Genoa, Italy, 449–454.

25. Doan, Anhai, Raghu Ramakrishnan, and Alon Y. Halevy. "Crowdsourcing systems on the world-wide web." Communications of the ACM 54.4 (2011): 86-96.

26. Dow, S., Kulkarni, A., Klemmer, S., Hartmann, B. Shepherding the Crowd Yields Better Work. *CSCW 2012*.

27. Erickson, T., C. Davis, W. Kellogg, and M. Helander. Assistance: The Work Practices of Human Administrative Assistants and their Implications for IT and Organizations. *Proc. CSCW 2008.* 2008.

28. Faulring, A., B. Myers, K. Mohnkern, B. Schmerl, A. Steinfeld, J. Zimmerman, A. Smailagic, J. Hansen, and D. Siewiorek. Agent-assisted task management that reduces email overload. *Proc. IUI 2010.* 2010.

29. Feldman, M.S. Constraints on Communication and Electronic Mail. *Proc. CSCW 1986.* 1986.

30. Fellbaum, C. Wordnet: *An Electronic Lexical Database*. MIT Press, Cambridge, MA, USA, 1998.

31. Fisher, D., A.J. Brush, E. Gleave, M.A. Smith. Revisiting Whittaker & Sidner's "email overload" ten years later. *Proc. CSCW 2006.* 2006.

32. Freed, M., J. Carbonell, G. Gordon, J. Hayes, B. Myers, D. Siewiorek, S. Smith, A. Steinfeld, and A. Tomasic. RADAR: A Personal Assistant that Learns to Reduce Email Overload. *In Proc. AAAI 2008*. 2008.

33. Galton, Francis. "Vox populi." Nature 75 (1907): 450-451.

34. Gick, M.L. and Holyoak, K.J. Analogical problem solving. *Cognitive psychology 12*, 3 (1980), 306-355.

35. Gollwitzer, P.M. The Psychology of Action: Linking Cognition and Motivation to Behavior. Guilford Press, New York, NY, USA, 1996.

36. Good, Benjamin M., and Andrew I. Su. "Games with a scientific purpose." Genome Biology 12.12 (2011): 135.

37. Gwizdka, J. TaskView – Design and Evaluation of a Task-based Email Interface. *Proc. CASCON 2002*. 2002.

38. Horowitz, D., Kamvar, S.D. The anatomy of a large-scale social search engine. WWW 2010, ACM Press, 431-440.

39. Horvitz, E., A. Jacobs, D. Hovel. Attention-sensitive alerting. *Proc. UAI 1999*. 1999.

40. Iqbal, S.T. and B.P. Bailey. Effects of Intelligent Notification Management on Users and Their Tasks. *CHI 2008*, pp.91-100.

41. Iqbal, S.T. and Bailey, B.P. Leveraging Characteristics of Task Structure to Predict the Cost of Interruption. *CHI 2006*, ACM Press, 741–750.

42. Jackson, T., R. Dawson, and D. Wilson. The Cost of Email Interruption. *Journal of Systems and Information Technology 5*, 1. 2001.

43. Janis, I.L. Groupthink: Psychological Studies of Policy Decisions and Fiascoes. Wadsworth Publishing, 1982.

44. Jones, W.P., Teevan, J. Personal Information Management. University of Washington Press, Seattle, WA, USA, 2007.

45. Karger, D., Oh, S., Shah, D. Iterative learning for reliable crowdsourcing systems. *NIPS 2011*, Granada, Spain.

46. Karpicke, J. D., Blunt, J. R., Retrieval practice produces more learning than elaborative studying with concept mapping. *Science* 331.68018 (2011), 772-775.

47. Kiritchenko, S. and Matwin, S. Email Classification with Co-Training. *Proc. CASCON 2001*. 2001.

48. Kittur, A., Chi, E., Pendleton, B. A., Suh, B., Mytkowicz, T. (2007). Power of the Few vs. Wisdom of the Crowd: Wikipedia and the Rise of the Bourgeoisie. Alt.CHI 2007.

49. Kittur, A., Nickerson, J., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M., Horton, J., *The Future of Crowd Work*. In Proc. CSCW 2013.

50. Kittur, A., Smus, B., Kraut, R. and Khamkar, S. Crowdforge: Crowdsourcing complex work. In *Proc. UIST 2011* pp. 43–52.

51. Kokkalis, N., Köhn, T., Huebner, J. , Lee, M., Schulze, F., Klemmer. S.R. TaskGenies: Automatically Providing Action Plans Helps People Complete Tasks. *ToCHI* 2013.

52. Kokkalis, N., Köhn, T., Pfeiffer, C., Chornyi, D., Bernstein, M., Klemmer, S.R. *EmailValet: Managing Email Overload through Private, Accountable Crowdsourcing*. In Proc. CSCW 2013.

53. Kulkarni, A.P., Can, M. and Hartmann, B. Turkomatic: automatic recursive task and workflow design for mechanical turk. *CHI EA 2011*, ACM Press, 2053–2058.

54. Kulkarni, C., Dow. S, and Klemmer, S.R., Early and Repeated Exposure to Examples Improves Creative Work. In Proc. *Cognitive Science*, 2012.

55. Lampert, A., R. Dale, C. Paris. Detecting Emails Containing Requests for Action. *Proc. ACL 2010*. 2010.

56. Law, E., von Ahn, L. Human computation. Synthesis Lectures on Artificial Intelligence and Machine Learning 5.3 (2011), 1-121.

57. Law, E., Zhang, H. Towards Large-Scale Collaborative Planning: Answering High-Level Search Queries Using Human Computation. In AAAI, San Francisco, 2011.

58. Lee, B., Srivastava, S., Kumar, R., Brafman, R. and Klemmer, S.R. Designing with interactive example galleries. *CHI 2010*, ACM Press (2010), 2257–2266.

59. Leventhal, H., Singer, R. and Jones, S. Effects of fear and specificity of recommendation upon attitudes and behavior. *Journal of Personality and Social Psychology 2*, 1 (1965), 20-29.

60. Lih, Andrew. The Wikipedia revolution: How a bunch of nobodies created the world's greatest encyclopedia. Hyperion, 2009.

61. Little, G., Chilton, L.B., Goldman, M. and Miller, R.C. *Exploring Iterative and Parallel Human Computation Processes*. In *Proc. KDD-HCOMP 2010*, ACM Press (2010).

62. Luszczynska, A. An implementation intentions intervention, the use of a planning strategy, and physical activity after myocardial infarction. *Social Science & Medicine 62*, 4 (2006), 900-908.

63. Mackay, W. E. More than Just a Communication System: Diversity in the Use of Electronic Mail. *Proc. CSCW 1988*. 1988.

64. Mark, G., Gudith, D., Klocke, U. The cost of interrupted work: more speed and stress. CHI 2008, ACM Press, 107-110.

65. Marsh, R.L., Landau, J.D. and Hicks, J.L. How examples may (and may not) constrain creativity. In *Memory & Cognition 24*, 5 (1996), 669-680.

66. Matejka, J., Grossman, T., Fitzmaurice, G. IP-QAT: In-Product Questions, Answers & Tips. In *Proc. UIST 2011*, ACM Press, 175–184.

67. Milkman, K., Beshears, J., Choi, J., Laibson, D., and Madrian, B. Using implementation intentions prompts to enhance influenza vaccination rates. In *Proc. National Academy of Sciences of the United States of America (PNAS) 108*, 26 (2011), 10415-10420.

68. Miller, G.A., WordNet: A Lexical Database for English. *Com. of the ACM*, November 1995, pp. 39-41.

69. Morris, M.R., Teevan, J., Panovich, K. What Do People Ask Their Social Networks, and Why? A Survey Study of Status Message Q&A Behavior. *CHI 2010,* ACM Press, 1739–1748.

70. Muller, M. J., and Gruen, D. M. Working Together Inside an Emailbox. *Proc. ECSCW 2005*. 2005.

71. New in Labs: Tasks. http://gmailblog.blogspot.com/2008/12/new-in-labs-tasks.html.

72. O'Donoghue, T., Rabin, M. Choice and Procrastination. *The Quarterly Journal of Economics 116*, 1 (Feb. 2001), 121-160.

73. Palen, L., and P. Dourish. Unpacking "Privacy" for a Networked World. *Proc. CHI 2003*. 2003.

74. Pirolli, P. and Card, S. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Proc. International Conference on Intelligence Analysis* (2005).

75. Quinn, Alexander J., and Benjamin B. Bederson. "Human computation: a survey and taxonomy of a growing field." Proceedings of the 2011 annual conference on Human factors in computing systems. ACM, 2011.

76. Rambow, O., L. Shrestha, J. Chen, and C. Lauridsen. Summarizing Email Threads. *Proc. HLT-NAACL 2004*. 2004.

77. Robinson, R., Calibrated Peer Review™. *The American Biology Teacher* 63.7 (2001), 474-480.

78. Sahami, M., S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian Approach to Filtering Junk E-mail. *AAAI Workshop: Learning for Text Categorization*. 1998.

79. Salvucci, D. D., & Bogunovich, P. (2010). Multitasking and monotasking: The effects of mental workload on deferred task interruptions. *Proc. CHI 2010*, 85-88.

80. Samiei, M., J. Dill, and A. Kirkpatrick. EzMail: Using Information Visualization Techniques to Help Manage Email. *Proc. IV 2004*. 2004.

81. Segal, R.B. and Kephart, J.O. MailCat: An Intelligent Assistant for Organizing E-Mail. *Proc. AAMAS 1999*. 1999.

82. Set up Mail Delegation. http://support.google.com/mail/bin/answer.py?hl=en&answer=138350.

83. Shen, J., L. Li, T.G. Dietterich, and J.L. Herlocker. A Hybrid Learning System for Recognizing User Tasks from Desktop Activities and Email Messages. *Proc. IUI 2006*. 2006.

84. Smith, S.M., Ward, T.B., Schumacher, J.S. Constraining effects of examples in a creative generation task. *Memory and Cognition 21*, 6 (1993), 837-845.

85. Strauss, A., Work and the Division of Labor. *The Sociological Quarterly*, 26(1), pp.1-19, 1985.

86. Surowiecki, J. The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies, and Nations. Doubleday, New York, NY, USA, 2004.

87. Takkinen, J. and N. Shahmehri. CAFE: A Conceptual Model for Managing Information in Electronic Mail. *Proc. HICSS 1998*. 1998.

88. Treynor, Jack L. "Market efficiency and the bean jar experiment." Financial Analysts Journal 43.3 (1987): 50-53.

89. Venolia, G., L. Dabbish, J.J. Cadiz, and A. Goopta. Supporting email workflow. *Technical Report MSR-TR-2001–88*. 2001.

90. von Ahn, L., Blum, M., Langford, J., Telling Humans and Computers Apart Automatically. *Com. of the ACM*, Feb. 2004.

91. von Ahn, L., Dabbish, L., Labeling images with a computer game. In Proc. CHI 2004, ACM Press, pp. 319-326.

92. Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. Science, September, 2008.

93. Vrandečić, D., Gil, Y., Ratnakar, Want World Domination? Win at Risk! *IUI 2011*, ACM Press, 323–326.

94. Wang, J., Ipeirotis, P.G., Provost, F. *Managing Crowdsourcing Workers*. Winter Conference on Business Intelligence, 2011.

95. Webb, Thomas L., and Paschal Sheeran. "Can implementation intentions help to overcome ego-depletion?." Journal of Experimental Social Psychology 39.3 (2003): 279-286.

96. Weber, R. The Grim Reaper: The Curse of E-Mail. *MIS Quarterly 28*, 3. 2004.

97. Weisberg, R. W. Creativity and Knowledge: A Challenge to Theories. In Sternberg, R.J., *Handbook of Creativity,* 226-50. Cambridge University Press, New York, NY, USA (1999).

98. Weiser, M. The Computer for the 21st Century. *Scientific American*. 1991.

99. Whittaker, S. and C. Sidner. Email Overload: Exploring Personal Information Management of Email. *Proc. CHI 1996*. 1996.

100. Winograd, T., and F. Flores. Understanding Computers and Cognition: A New Foundation for Design. Ablex, 1986.

101. Yorke-Smith, N., Saadati, S., Myers, K., Morley, D. *Like an Intuitive and Courteous Butler: A Proactive Personal Agent for Task Management*. SRI's Artificial Intelligence Center, 2009.

102. Zeigarnik, B. *Das Behalten von erledigten und unerledigten Handlungen*. In Untersuchungen zur Handlungs- und Affektpsychologie. Psychologische Forschung, 9, 1-85, 1927.

103. Zhang, H., Law, E., Gajos, K., Horvitz, E., Miller, R., Parkes, D. *Human Computation Tasks with Global Constraints.* In CHI 2012, Austin.