TaskGenies: Automatically Providing Action Plans Helps People Complete Tasks

NICOLAS KOKKALIS, THOMAS KÖHN, JOHANNES HUEBNER, MOONTAE LEE, FLORIAN SCHULZE, and SCOTT R. KLEMMER, Stanford University

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 article demonstrates that these benefits can also be realized when these plans are created by others or reused from similar tasks. Four experiments test these approaches, finding that people indeed complete more tasks when they receive externally-created action plans. To automatically provide plans, we introduce 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. We demonstrate that it is possible for people to create action plans for others, and we show that it can be cost effective.

Categories and Subject Descriptors: H.4.1 [**Information Systems Applications**]: Office Automation—*Time management*

General Terms: Design, Human Factors, Experimentation

Additional Key Words and Phrases: Task management, Crowdsourcing, action plans, implementation intentions, time management

ACM Reference Format:
Kokkalis, N., Köhn, T., Huebner, J., Lee, M., Schulze, F., and Klemmer, S. R. 2013. TaskGenies: Automatically providing action plans helps people complete tasks. *ACM Trans. Comput.-Hum. Interact.* 20, 5, Article 27 (November 2013), 25 pages.
DOI: http://dx.doi.org/10.1145/2513560

1. INTRODUCTION

People complete tasks faster when they develop concrete implementation intentions [Allen 2002; Amabile and Kramer 2011; Gollwitzer 1996; Leventhal et al. 1965; Luszczynska 2006; Milkman et al. 2011]. Several controlled experiments have found that people assigned to make concrete plans follow through more often – from getting flu shots [Milkman et al. 2011] to exercising for heart attack recovery [Luszczynska 2006] – than those only required to formulate a high-level theory. This benefit could arise from the *availability* of an action plan (regardless of source) and/or the process of *contemplating* a plan oneself. This work seeks to disambiguate these possibilities.

We introduce and evaluate crowdsourcing and community approaches for creating plans, and NLP techniques for reusing them. A between-subjects experiment found

S. R. Klemmer is also affiliated with the University of California, San Diego. His email at UCSD is srk@ucsd.edu.
This research was sponsored in part by NSF POMI 2020 Grant No. CNS-0832820.
Authors' address: N. Kokkalis, T. Köhn, J. Huebner, M. Lee, F. Schulze, and S. R. Klemmer, Stanford University, HCI Group, Computer Science Department, 353 Serra Mall, Stanford, CA 94305; email: nicolas@cs.stanford.edu.

DOI: http://dx.doi.org/10.1145/2513560

that people receiving crowd-created plans completed more tasks than people prompted to create their own plans and than 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 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 that combines 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.

To help workers efficiently produce good plans, TaskGenies strategically shows them examples of prior work. Its approach seeks to address a tention with using examples. While, viewing others' high-quality work can increase performance by norming expectations to a high level [Bandura and McClelland 1977], viewing existing answers to a *current* task risks lazy copying and/or priming-induced conformity [Smith et al. 1993].

To balance this tension, the Genies workflow employs 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.

2. 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.

2.1. Auto-Provided Plans Increase Task Completion Rate

Is it realistic to ask crowdsourced workers to provide action plans? We hypothesize 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 article 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.

2.2. Action Plans Differentially Benefit Different Task Types

When tasks are daunting, complex, and/or novel, an action plan may help 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 might not be an 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.

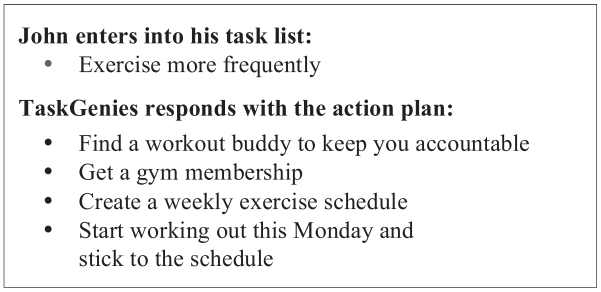

Fig. 1. Decomposing people's tasks to concrete steps (action plans) makes high-level tasks more actionable. This way, tasks linger less and people complete more of them. Online crowds create new plans; algorithms identify and reuse existing ones.

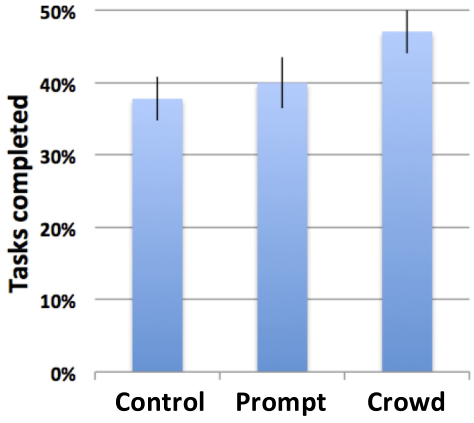

Fig. 2. Participants in the crowd completed significantly more tasks than those in the control and prompt conditions. Error bars indicate 95% CI.

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. We 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 2). 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.

2.3. 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.

2.4. 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 article 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 compared to 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.

2.5. Contributions

This article provides the following contributions.

— It introduces the Genies crowdsourcing method, which increases quality by strategically presenting examples.

— It demonstrates that action plans help people complete more tasks, especially with high-level and lingering to-dos. These action plans are helpful even when reused across different people with similar tasks.
— It shows that crowdsourcing can be an effective way to create action plans, when used with the Genies pattern and TaskGenies system.
— It introduces a technique for successfully scaling this approach through a combination of community-produced action plans and NLP-based reuse.
— It demonstrates the first task management system with automatically provided action plans, and shows both its benefits and practicality.

3. RELATED WORK

3.1. What Makes Action Plans Effective?

Popular and academic writing on work emphasizes that formulating actionable steps benefits both efficiency [Allen 2002] and morale [Amabile and Kramer 2011]. 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 [O'Donoghue et al. 2001].

Moreover, having concrete, separated steps provides people with helpful guides on when to suspend work, especially to handle interruptions [Iqbal and Bailey 2006]. 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 [Iqbal and Bailey 2008; Salvucci and Bogunovich 2010]. Despite these benefits, people often fail to plan because the costs are immediate but the benefits are deferred [Allen 2002; Bellotti et al. 2003].

With complex activities, the articulation work of planning and organizing tasks comprises an important piece of the job [Bellotti et al. 2004; Strauss 1985]. 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 not help much. Moreover, the way people record tasks may require contextual knowledge 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.

3.2. Approaches for Organizing Crowd Work

This work introduces a crowd-creation strategy for providing action plans. Crowdsourcing is most valuable when algorithms alone can't provide sufficient results [von Ahn and Dabbish 2004]. However, the cost and difficulty of acquiring labor presents a bottleneck to broader crowdsourcing use. Consequently, we also introduce two strategies for scaling plan provisioning: *community creation* and *algorithmic reuse*. Here, we review work related to these three strategies.

3.2.1. Crowd Creation. The literature offers several approaches for organizing crowd workers to solve open-ended problems [Kittur et al. 2013; Law and von Ahn 2011]. The simplest way to allocate workers is to assign exactly one worker per task. However, not all crowd workers are diligent and/or competent.

To increase quality, multiple workers can be redundantly assigned the same task. Figure 3 sketches alternative approaches. Worker solutions can be averaged, computed by pairwise agreement [von Ahn and Dabbish 2004], simple majority vote [von Ahn et al. 2004, 2008], or weighted by estimated worker quality [Dai et al. 2011; Karger

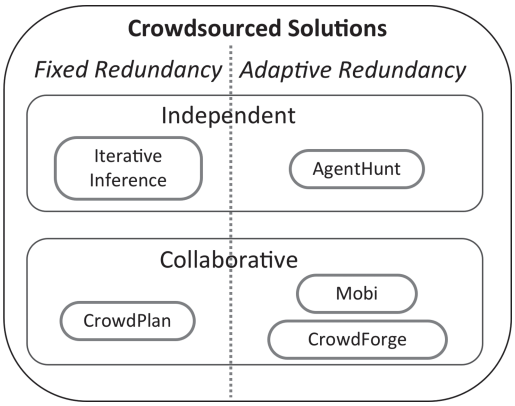

Fig. 3. A sketch of different approaches for assigning workers to tasks, showing exemplars of key alternatives. Quality can be improved through redundancy, refinement, or hierarchical approaches using a *fixed* or an *adaptive* work redundancy.

et al. 2011]. This *independent redundancy* parallelizes naturally and, under certain conditions, yields higher quality [Surowiecki 2004].

The number of workers to assign to tasks can even be decided *adaptively* (e.g., Dai et al. [2011]). For more sophisticated tasks, workers can *collaboratively refine* others' work (e.g., Little et al. [2010], Zhang et al. [2012], and Kittur et al. [2011]) or be organized into *multiple roles* (e.g., Bernstein et al. [2010], Kittur et al. [2007], Kulkarni et al. [2012], and Little et al. [2010]). 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 [Bernstein et al. 2010; Law and Zhang 2011]). More complex approaches delineate a separate phase for merging and refining different ideas [Ahmad et al. 2011; Kittur et al. 2007]. Other techniques provide workers with more flexible workflow control to choose what they do (do work, refine, assess) [Zhang et al. 2012]. The appropriateness and relative merits of these different approaches are being actively studied.

Closest in domain to this article, CrowdPlan [Law and Zhang 2011] 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 [Zhang et al. 2012], others use them to indicate what not to produce [Law and Zhang 2011], still others do not use any examples.

3.2.2. Community Creation. To recruit labor, it can be effective to leverage the user community and user's social networks. Routing tasks to peers can be handled by self-declared expertise [Ackerman and Malone 1990], by publically posting to an online forum [Ackerman and Malone 1990; Matejka et al. 2011], by posting to one's social network [Bernstein et al. 2009; Morris et al. 2010], or automatically based on inferred expertise and availability [Horowitz and Kamvar 2010]. When planners help each other, it helps the system scale and can engender a community ethos.

3.2.3. Automatic Reuse. Interfaces for identifying and presenting relevant examples can enable reuse. People can actively search for such examples [Brandt et al. 2010],

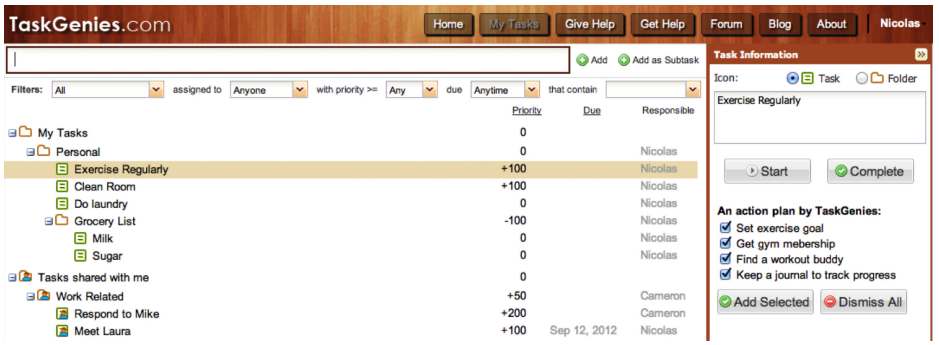

Fig. 4. TaskGenies Web interface. Users can add steps of a provided action plan (right) as sub-tasks (left).

or they can be automatically provided [Vrandečić et al. 2011; Wang et al. 2011]. 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 [Yorke-Smith et al. 2009].

4. 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.

4.1. Multiple Convenient Ways to Enter Tasks

TaskGenies has three interfaces for submitting tasks. First, a user can send a task by e-mail 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 (Figures 4 and 5). Users can request plans for tasks, and have the option to auto-request plans when a task lingers for several days.

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.

4.2. 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 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. 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 4). Users of my.tasksgenies.com can resubmit the task to the crowd, providing clarifications, if they don't like the action plan they received.

4.3. NLP Identifies Similar Tasks to Reuse Action Plans

We created the GeNiLP Natural Language Processing technique to identify similar tasks and reuse action plans. This technique, given a corpus of existing tasks C and a new task t, outputs the most similar task $s \in C$ and a similarity coefficient $c \in [0,1]$. 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.

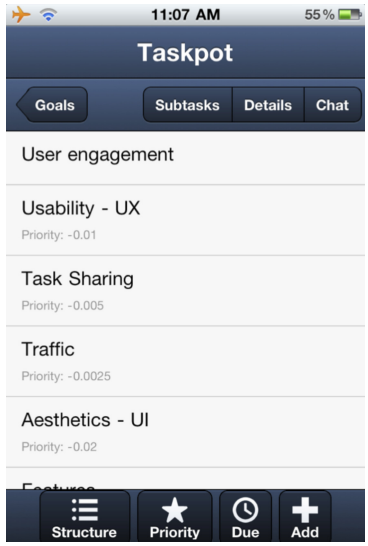

Fig. 5. TaskGenies mobile task list.

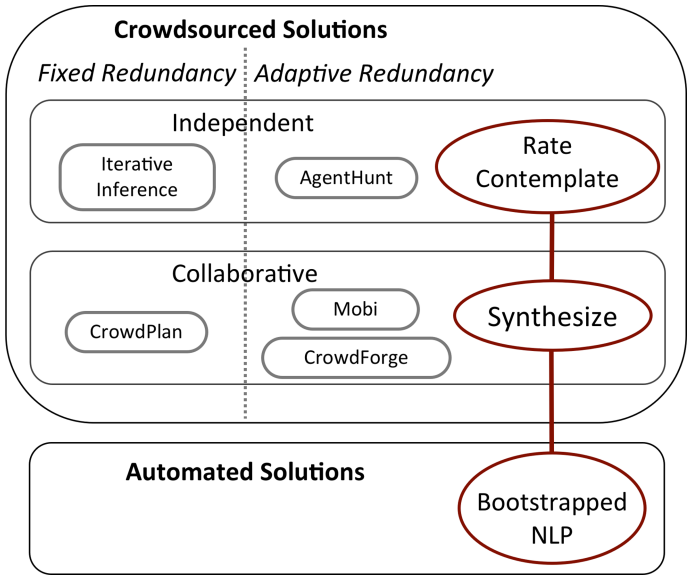

Fig. 6. Genies crowdsources novel solutions. Rating and contemplating happen independently, whereas "synthesizing" collaboratively refines the final outcome. Existing solutions can be reused through NLP.

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 [Miller 1995] to match tasks with similar meaning (e.g., "buy pants" and "purchase trousers"; or "Email violin teacher" and "Call piano teacher to schedule lessons").

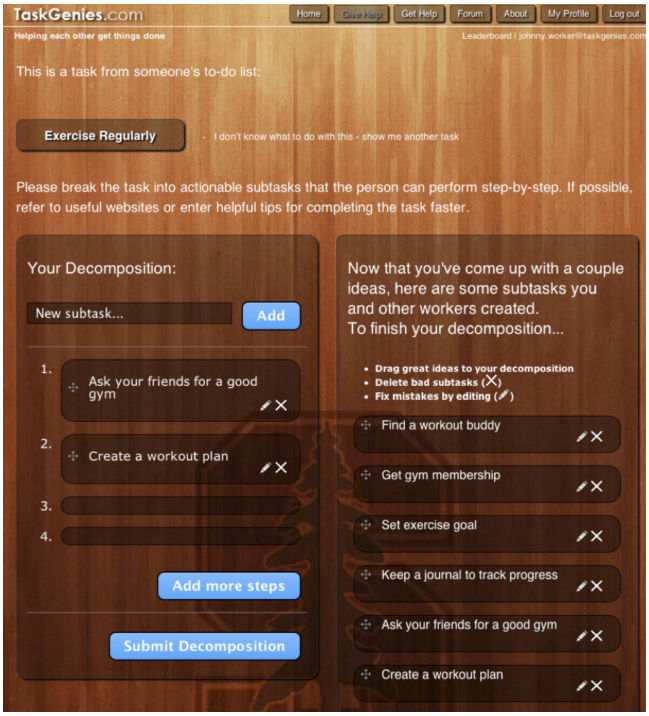

Fig. 7. Workers use this interface to create action plans. After entering a couple of steps the system shows relevant examples on the right.

5. 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 is composed of three steps (Figure 6).

(1) *Rate*. Taking a cue from calibrated peer assessment [Robinson 2001], Genies begins with a calibration step where workers view and rate results of different but analogous tasks. This serves several functions simultaneously: the ratings find good solutions; train and calibrate 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 task or similar, integrating elements from them when useful (see Figure 7).
(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 [Dow et al. 2012; Kulkarni et al. 2012; Lee et al. 2010; Marsh et al. 1996].

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. (Just the studies for this article

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 five ratings, converging on an accepted plan in an average of 2.2 workers per task.

5.1. 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. 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., Law and Zhang [2011]), 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, Law and Zhang [2011] 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 7). The worker is encouraged to review, drag-and-drop, and 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 $\times$ button; this lowers that step's ranking in the future.

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).

6. STUDY 1: CROWD-CREATED ACTION PLANS

6.1. 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 e-mail 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 website. Instead, participants submitted tasks through a generic Web form.

6.1.1. Participants. 280 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.

6.1.2. 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.

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 e-mail. 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).

Prompt. Ninety-three participants were sent e-mails asking them to create action plans for their own tasks. To make this condition parallel to the crowd condition, each e-mail listed one task, asked participants to create an action plan for it, and suggested: "Before you close this e-mail, could you possibly start this task by completing its first applicable step right away?" To ensure that the number and timing of e-mails was consistent with the crowd condition, each participant was randomly paired with a participant in the Crowd condition. Whenever TaskGenies e-mailed that Crowd participant an action plan, TaskGenies also e-mailed the corresponding Prompt participant. The wording and structure of these e-mails was designed to be as similar as possible. See Appendix B for such an e-mail. 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.

6.2. Dependent Measures

6.2.1. 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 e-mail 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. 82 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 paper 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.

6.2.2. 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).

6.2.3. Lingering Tasks Completed. The analysis included all tasks with completion information excluding tasks where participants did not respond about completion. 82

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

Fig. 8. Each column shows one participants 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.

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).

6.2.4. Understandable Tasks Completed. 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).

6.3. Results

6.3.1. 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 ($F(2,225) = 4.85$, $p < 0.01$). Follow-up pairwise t-tests with false discovery rate (FDR) correction found a significant difference between Crowd and Control ($t = 3.07$, $p < 0.01$), and between Crowd and Prompt ($t = 2.19$, $p < 0.05$). No difference was found between Prompt and Control ($t = 0.69$, $p > 0.05$). The results are graphically presented in Figure 2. Figure 8 shows the task list for an example participant in each condition. Figure 9 shows an example of a crowd-created action plan.

6.3.2. 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; $\chi^2 = 16.00$, $p < 0.0001$). Participants completed 1.13 times more well-defined tasks with an action plan. (44.9% of 639 in Crowd vs.

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.

Fig. 9. Crowd-created action plans can provide valuable information. However, this does not guarantee completion. See more action plans on taskgenies.com.

39.6% of 712 in Control; $\chi^2 = 3.68, p = 0.055$.) This suggests that crowd-created action plans have a larger effect for high-level tasks than well-defined ones.

6.3.3. Lingering Tasks. Providing action plans significantly increased task completion rates for both lingering and nonlingering 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 nonlingering 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 nonlingering ones.

6.3.4. Comprehendible Tasks. We measure 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 ($R = 0.367$).

7. 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.

7.1. Method

7.1.1. Participants. 388 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.

7.1.2. 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 e-mail encouraged the recipient to visit TaskGenies and create plans for others. The rest of the method was the same as Study 1.

7.2. Dependent Measures

7.2.1. 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 e-mail with a link to a Web page listing their 20 tasks. Participants were instructed to mark the tasks they completed.

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

7.3. Results

7.3.1. 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 ($t = 2.18, p < 0.05$).

7.3.2. 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 10 depicts the distribution of contribution among participants.

7.3.3. 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 11).

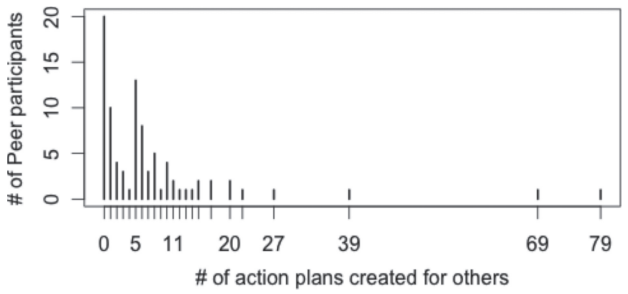

Fig. 10. The study ensured that everyone in the Community condition received 20 action plans each. However, participants contributed at very different levels.

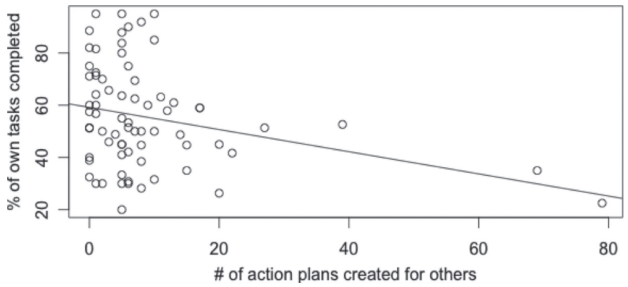

Fig. 11. The more Community participants created action plans for others, the fewer tasks they completed themselves.

8. STUDY 3: RECYCLING ACTION PLANS

Studies 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, we used the GeNiLP algorithm.

8.1. Method

As a corpus, we 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 Studies 1 and 2 and other actual users of TaskGenies. Using this algorithm, we conducted an experiment similar to Study 1 with two conditions: Recycle and Control.

8.1.1. Participants. 39 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.

8.1.2. Procedure. As in Study 1, participants provided 10 lingering and 10 nonlingering tasks they were planning to complete in the near future. Participants across all conditions were unaware of the TaskGenies website and received action plans by e-mail.

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, we 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, we sent the action plans by e-mail and each e-mail listed one task. To ensure that the number and timing of e-mails 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 e-mails was the same as in Crowd condition. See Appendix A for such an e-mail. The rest of the method was the same as in Study 1.

8.2. 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 e-mail 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.

8.3. Results

8.3.1. 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.

8.3.2. Qualitative Analysis. Both the strengths and weaknesses of GeNiLP (see Appendix D) come from its synonym-engine WordNet [Fellbaum 1998]. 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 multiword 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.

9. STUDY 4: COMPARING GENIES WITH OTHER APPROACHES

A study compared Genies to three other workflows by creating action plans for the same 10 tasks. We 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 (the second author) rated all plans on a 10-point scale of perceived quality. The four workflows were as follows.

(1) *Parallel.* Workers independently create action plans.

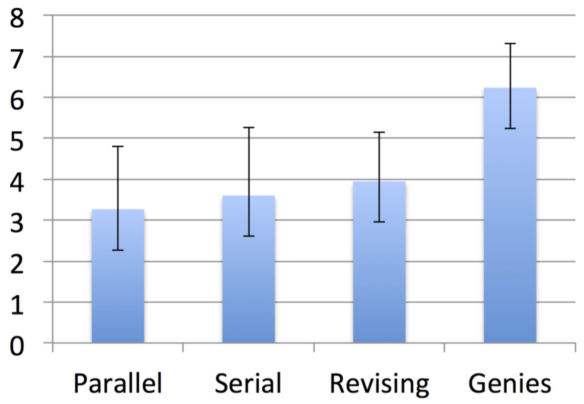

Fig. 12. Average action plan ratings for each workflow on a 10-point scale.

(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.

9.1. Results

The overall rating for Genies action plans was higher than other conditions (see Figure 12). 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 ($F(3,36) = 4.30$, $p < 0.05$). Follow-up t-tests with false discovery rate (FDR) correction found a significant difference between Genies and Parallel ($p < 0.01$), between Genies and Serial ($p < 0.05$), and between Genies and Revising ($p < 0.0001$). No difference was found between the other conditions.

9.1.1. 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 [Karpicke and Blunt 2011]. 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.

9.1.2. 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

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.

Fig. 13. 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.

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 13 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 [Lee et al. 2010; Marsh et al. 1996; Weisberg 1999], sometimes constrain it [Smith et al. 1993], and sometimes are surprisingly underused [Gick and Holyoak 1980]. 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 [Kulkarni et al. 2012].

10. 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.

10.1. 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 [Ahmad et al. 2011; Amabile and Kramer 2011].

We see three reasons why crowd-created plans improved completion more for lingering tasks than nonlingering 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.

We should also point out that, sometimes, deferring tasks can be a savvy choice to focus on more important or urgent work [Bellotti et al. 2004] 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.

10.1.1. 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.

10.2. 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, for example, Wikipedia [Kittur et al. 2007]. 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.

10.3. 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 [Law and Zhang 2011]. 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 a wisdom of the crowd [Surowiecki 2004]. 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 [Lee et al. 2010; Marsh et al. 1996], even if it sometimes reduces independence or increases groupthink [Smith et al. 1993]. 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 [Law and Zhang 2011]. 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. The results of Kulkarni et al. [2012] support this intuition.

10.4. 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.

10.5. The Genies Pattern Benefits and Limitations

Genies train 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? Do 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.

10.6. 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, Cardmunch[1] 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

[1]http://www.cardmunch.com

tasks in the database users can receive an action plan, without sharing the task with another person. For novel tasks, users can elect whether or not to share them and receive benefits.

11. CONCLUSIONS AND FUTURE WORK

This article 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. We created TaskGenies, a novel system that supported our experiments, and showed that crowdsourcing is an effective way to create action plans. To scale plan provision, we introduced a community approach where users create tasks for each other. Further scaling was accomplished by combining human creation with algorithmic reuse. We 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.

E-mail is the de facto task list for many people – where unread, flagged, or starred messages signify tasks. However, the e-mail interface of sender names and subject line headers is poorly suited to managing tasks. Task list systems may be powerfully integrated into e-mail, and crowd, community, and algorithmic approaches can help transform e-mails into actionable tasks [Kokkalis et al. 2013].

In this article, crowd-provided plans worked best when they required little contextual knowledge. Algorithms or crowd workers could elicit additional contextual information when necessary. Context-aware routing – using a social network [Horowitz and Kamvar 2010], 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?

Finally, 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 friends, peers or 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 friends and family.

APPENDIXES

Appendix A. E-mail 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 e-mail, 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. E-mail 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 e-mail, 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. E-mail 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 e-mail, 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.

ALGORITHM 1. Overall NLP 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 [de Marneffe et al. 2006] to identify all modifier words $M(w)$ of w within t. (ii) Use WordNet [Fellbaum 1998; Miller 1995] to find each sense $s_{w,i}$ of w, (iii) For each $s_{w,i}$ compare the definitions and examples of $s_{w,i}$ with the definitions and examples of all senses of the words in $M(w)$ and count the number of agreements, (iv) Select the sense $s_{w,i}$ 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.

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 here.

Great Matches
Study for 15-122 midterm → study for midterm
Buy groceries. → buy groceries
Buy tickets to New York → Buy flight tickets

ALGORITHM 2. Similarity Computation *(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
```

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

ACKNOWLEDGMENTS

The authors thank Steven Diamond, Michael Chang, and Dominic R. Becker for their contributions during their Summer internship.

REFERENCES

Ackerman, M. A. and Malone, T. W. 1990. Answer garden: A tool for growing organizational memory. In *Proceedings of SIGOIS*. MIT, 31–39.

Ahmad, S., Battle, A., Malkani, Z., and Kamvar, S. D. 2011. The Jabberwocky programming environment for structured social computing. In *Proceedings of UIST*.

Allen, D. 2002. *Getting Things Done: The Art of Stress-Free Productivity*. Penguin, New York.

Amabile, T. and Kramer, S. 2011. *The Progress Principle.* Harvard Business Review Press, Boston, MA.

Bandura, A. and McClelland, D. C. 1977. *Social Learning Theory*. Prentice-Hall, Englewood Cliffs, N.J.

Bellotti, V., Ducheneaut, N., Howard, M., et al. 2003. Taking email to task: The design and evaluation of a task management centered email tool. In *Proceedings of CHI*. ACM, New York, 345–352.

Bellotti, V., Dalal, B., Good, N., et al. 2004. What a to-do: Studies of task management towards the design of a personal task list manager. In *Proceedings of CHI*. ACM Press, 735–742.

Bernstein, M., Tan, D., Smith, G., et al. 2009. Collabio: A game for annotating people within social networks. In *Proceedings of UIST*. ACM, New York, 97–100.

Bernstein, M., Little, G., Miller, R. C., et al. 2010. Soylent: A word processor with a crowd inside. In *Proceedings of UIST*. ACM, 313–322.

Brandt, J., Dontcheva, M., Weskamp, M., and Klemmer, S. R. 2010. Example-centric programming: Integrating web search into the development environment. In *Proceedings of CHI*. 513–522.

Dai, P., Mausam, and Weld, D. S. 2011. Artificial intelligence for artificial artificial intelligence. In *Proceedings of AAAI*.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*. 449–454.

Dow, S., Kulkarni, A., Klemmer, S., and Hartmann, B. 2012. Shepherding the crowd yields better work. In *Proceedings of CSCW*.

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

Gick, M. L. and Holyoak, K. J. 1980. Analogical problem solving. *Cognit. Psych. 12*, 3, 306–355.

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

Horowitz, D. and Kamvar, S. D. 2010. The anatomy of a large-scale social search engine. In *Proceedings of WWW*. ACM, 431–440.

Iqbal, S. T. and Bailey, B. P. 2006. Leveraging characteristics of task structure to predict the cost of interruption. In *Proceedings of CHI*. ACM Press, 741–750.

Iqbal, S. T. and Bailey, B. P. 2008. Effects of intelligent notification management on users and their tasks. In *Proceedings of CHI*. 91–100.

Karger, D., Oh, S., and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. In *Proceedings of NIPS*.

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

Kittur, A., Chi, E., Pendleton, B. A., Suh, B., and Mytkowicz, T. 2007. Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. In *Proceedings of CHI*.

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

Kittur, A., Nickerson, J., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M., and Horton, J. 2013. The future of crowd work. In *Proceedings of CSCW*.

Kokkalis, N., Kohn, T., Pfeiffer, C., Chornyi, D., Bernstein, M., and Klemmer, S. 2013. EmailValet: Managing email overload through private, accountable crowdsourcing. In *Proceedings of CSCW.*

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

Kulkarni, C., Dow, S., and Klemmer, S. R. 2012. Early and repeated exposure to examples improves creative work. In *Proceedings of Cognitive Science*.

Law, E. and Zhang, H. 2011. Towards large-scale collaborative planning: Answering high-level search queries using human computation. In *Proceedings of AAAI*.

Law, E. and von Ahn, L. 2011. Human computation. *Synth. Lect. Artifi. Intell. Mach. Learn. 5*, 3, 1–121.

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

Leventhal, H., Singer, R., and Jones, S. 1965. Effects of fear and specificity of recommendation upon attitudes and behavior. *J. Personal. Soc. Psych. 2*, 1, 20–29.

Little, G., Chilton, L. B., Goldman, M., and Miller, R. C. 2010. Exploring iterative and parallel human computation processes. In *Proceedings of KDD-HCOMP*. ACM.

Luszczynska, A. 2006. An implementation intentions intervention, the use of a planning strategy, and physical activity after myocardial infarction. *Soc. Sci. Med. 62*, 4, 900–908.

Marsh, R. L., Landau, J. D., and Hicks, J. L. 1996. How examples may (and may not) constrain creativity. *Mem. Cognit. 24*, 5, 669–680.

Matejka, J., Grossman, T., and Fitzmaurice, G. 2011. IP-QAT: In-product questions, answers & tips. In *Proceedings of UIST*. ACM, 175–184.

Milkman, K., Beshears, J., Choi, J., Laibson, D., and Madrian, B. 2011. Using implementation intentions prompts to enhance influenza vaccination rates. *Proc. National Acad. Sci. 108*, 26, 10415–10420.

Miller, G. A. 1995. WordNet: A lexical database for English. *Comm. ACM*, 39–41.

Morris, M. R., Teevan, J., and Panovich, K. 2010. What do people ask their social networks, and why? A survey study of status message Q&A behavior. In *Proceedings of CHI*. ACM, 1739–1748.

O'Donoghue, T. and Rabin, M. 2001. Choice and procrastination. *Quart. J. Econom. 116*, 1, 121–160.

Robinson, R. 2001. Calibrated peer review™. *The American Biology Teacher* 63.7, 474–480.

Salvucci, D. D. and Bogunovich, P. 2010. Multitasking and monotasking: The effects of mental workload on deferred task interruptions. In *Proceedings of CHI*. 85–88.

Smith, S. M., Ward, T. B., and Schumacher, J. S. 1993. Constraining effects of examples in a creative generation task. *Mem. Cognit. 21*, 6, 837–845.

Strauss, A. 1985. Work and the division of labor. *Sociological Quart. 26*, 1, 1–19.

Surowiecki, J. 2004. 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.

von Ahn, L. and Dabbish, L. 2004. Labeling images with a computer game. In *Proceedings of CHI*. ACM Press, 319–326.

von Ahn, L., Blum, M., and Langford, J. 2004. Telling humans and computers apart automatically. *Comm. ACM*.

von Ahn, L., Maurer, B., McMillen, C., Abraham, D., and Blum, M. 2008. reCAPTCHA: Human-based character recognition via web security measures. *Science*.

Vrandečić, D., Gil, Y., and Ratnakar, V. 2011. Want world domination? Win at risk! In *Proceedings of IUI*. ACM, 323–326.

Wang, J., Ipeirotis, P. G., and Provost, F. 2011. Managing crowdsourcing workers. In *Proceedings of Winter Conference on Business Intelligence*.

Weisberg, R. W. 1999. Creativity and knowledge: A challenge to theories. In *Handbook of Creativity*, Cambridge University Press. Cambridge, UK, 226–250.

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

Zhang, H., Law, E., Gajos, K., Horvitz, E., Miller, R., and Parkes, D. 2012. Human computation tasks with global constraints. In *Proceedings of CHI*.

Received June 2012; revised November 2012, February 2013; accepted March 2013