Mechanical Novel: Crowdsourcing Complex Work through Reflection and Revision

Joy Kim, Sarah Sterman, Allegra Argent Beal Cohen, Michael S. Bernstein
Stanford University
{jojo808, ssterman, aacohen}@stanford.edu, msb@cs.stanford.edu

ABSTRACT
Crowdsourcing systems accomplish large tasks with scale and speed by breaking work down into independent parts. However, many types of complex creative work, such as fiction writing, have remained out of reach for crowds because work is tightly interdependent: changing one part of a story may trigger changes to the overall plot and vice versa. Taking inspiration from how expert authors write, we propose a technique for achieving interdependent complex goals with crowds. With this technique, the crowd loops between reflection, to select a high-level goal, and revision, to decompose that goal into low-level, actionable tasks. We embody this approach in Mechanical Novel, a system that crowdsources short fiction stories on Amazon Mechanical Turk. In a field experiment, Mechanical Novel resulted in higher-quality stories than an iterative crowdsourcing workflow. Our findings suggest that orienting crowd work around high-level goals may enable workers to coordinate their effort to accomplish complex work.

Author Keywords
Social computing; online creative collaboration; crowdsourcing; storytelling.

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Collaborative computing

INTRODUCTION

I know very dimly when I start what’s going to happen. I just have a very general idea, and then the thing develops as I write. —Aldous Huxley [12]

Crowdsourcing platforms such as Amazon Mechanical Turk bring together tens to thousands of people to accomplish complex work at massive scale, allowing the crowd to collaborate on goals such as researching purchases [19], classification tasks [35], and even creating music videos [20]. Currently, crowdsourcing systems accomplish these types of large tasks by decomposing work into independent microtasks. These microtask systems present work in an assembly line-like structure called a workflow [6], using mechanisms such as iteration [25], clustering [9], voting [26], and other patterns for splitting work [5, 19, 21]. Because these microtasks are independent, crowd workers can complete work without worrying about how their contributions affect others. As a result, large goals can be achieved quickly and at scale.

However, effective workflows are difficult to create in advance. To design a workflow of microtasks, an expert must first form a well-defined problem, then engage in an expensive and time-consuming process where they repeatedly test and iterate on potential workflow designs. Furthermore, the expert may run into common problem-solving barriers such as design fixation [15], difficulty decomposing work into microtasks [17], and fear of failure [4]. This process is difficult for crowds as well: systems like CrowdForge [19] and Turkomatic [21] have explored how the crowd can dynamically help experts decide how to partition work, but have found that workers require expert intervention [21] or a high-level initial decomposition of tasks [19] in order to decompose work without derailing from the intended goal.

In contrast, skilled creators iteratively create and revise goals to develop their vision as they work [11, 34]. That is, they know that problems are not always well-defined and that they may need to make many attempts before a solution becomes clear. With this in mind, we introduce a technique for continually updating and executing high-level goals with crowds. Rather than asking the crowd to help decompose a static goal, this technique loops between two phases: reflecting on the crowd’s progress so far to brainstorm and choose a high-level goal, and revising the artifact by decomposing that goal into actionable, low-level tasks through which workers make edits. For example, crowdworkers writing a short story could decide that a story ends too abruptly, and act on that in a specific way by brainstorming a different ending. This new goal can guide workers in deciding how other parts of the story need to change and unlock appropriate parts of the story for editing. Each goal can still be decomposed into microtasks, making this approach usable in existing crowdsourcing environments.

We instantiate our crowdsourcing strategy of reflection and revision in Mechanical Novel, a system that coordinates crowd workers from Amazon Mechanical Turk to write short fiction stories. Fiction writing was chosen as a test domain.
Figure 1. Mechanical Novel’s crowdsourcing loop alternates between high-level reflection to set a goal, and low-level revision to execute that goal.

due to the difficulty of defining clear expected solutions (i.e., many different types of stories are acceptable instantiations of an initial idea) and its inherent resistance to being broken down into independent subtasks. For this reason, collaboratively writing high-quality stories has been repeatedly explored by previous work [28, 1, 17, 18] but has remained out of reach for crowds without the help of a leader.

In Mechanical Novel, after first creating an initial first draft of a story based on a story prompt, workers select a goal by reflecting on their progress on the work so far: workers generate critiques, which includes suggesting a possible direction for how the story could change (e.g., foreshadow the death of a love interest). After voting among these suggestions to choose the next high-level goal to work towards, workers then execute the goal. Workers select which parts of the story need to change in order to address the high-level goal, and suggest a specific change for each part of the story they selected (e.g., the love interest says, “I will always be here for you”). This decomposes the high-level goal into specific tasks the crowd can act on. Workers then vote on these low-level suggestions, and revise portions of the story based on these tasks. The process then repeats, allowing the crowd to further improve the story by selecting a new goal to pursue.

In a controlled study comparing an iterative crowdsourcing workflow with Mechanical Novel, Mechanical Novel produced stronger stories as rated by readers. Specifically, Mechanical Novel’s stories had stronger plots (with clearer beginnings, middles, and ends). In iterations on six story drafts with known narrative problems, Mechanical Novel identified and successfully fixed high-level problems with plot and character, in contrast to the iterative workflow’s focus on spelling and grammar.

In summary, this paper makes the following contributions:

- The reflect and revise crowdsourcing technique, which enables crowds to collectively monitor their progress and flexibly contribute work based on high-level goals of their choosing.
- Mechanical Novel, an example system that demonstrates this technique in the context of storywriting, a domain that has typically remained out of reach for crowdsourcing systems.
- An evaluation of Mechanical Novel that shows the reflect and revise technique can generate short stories with stronger high-level characteristics (such as plot and character) than stories generated by a control system.

Crowds that are able to collectively articulate and execute high-level goals as they work could enable not just collaborative fiction-writing but a new class of crowd-powered work, including breaking news stories that are revised in real-time as new information appears, or reworking films across several stages or mediums (e.g., from a script to a storyboard to video).

RELATED WORK

We focus on developing techniques that allow the crowd to select and act on high-level goals. To inform our design, we examine the strengths and limitations of how crowds work together in existing collaborative environments.

Collaborating through context-free tasks

People often divide collaborative writing work by identifying sections of text that are independent from each other, and then working in parallel on a single document or writing in turn [16, 30]. Many crowdsourcing strategies think about tasks in a similar manner. In these, the role of subtasks is to produce sub-results that are mergable into the final result: crowdworkers caption sections of a speech by captioning one small snippet at a time [22]; flash teams frame collaborative expert crowd work around sequences of linked tasks and finding appropriate inputs and outputs from one phase to another [32]; workers create a music video by drawing one video frame at a time [20]; and still other work propose patterns [5, 19, 21] for breaking down complex tasks into context-free subtasks. These workflows can often produce complex work more quickly or more accurately than a person working alone.

Another approach is iterative crowdsourcing, where, rather than stopping after a result is put together piece by piece, one worker creates a first draft of the task, and later workers improve it with subsequent tasks. This is already visible in wiki and open source collaborations, where contributors base their own work on work by others. In tasks such as writing factual descriptions, transcribing blurry text, and brainstorming, iterative crowdsourcing processes can improve the quality of work over time [25].

At the same time, these workflows are fragile because they cannot flexibly react to change. Results put together piece-by-piece or in parallel may not be coherent, and iterative processes may fixate on improving low-quality work rather than
restoring to find a stronger concept [25]. Similar problems can be seen in existing collaborative storytelling platforms online, which are often implementations of round-robin storytelling games [1] that do not allow contributors to alter work that has previously been submitted; a new character introduced on a whim by one contributor unilaterally affects all later contributions whether it is good for the story or not. In other words, workflows lack support for reciprocal interdependence [37], where changing one part of the work may necessitate changes to other parts at any time. Mechanical Novel, instead, supports reciprocal interdependence by allowing workers to revisit and amend the high-level goals toward which they’re working.

Crowdsourcing with global goals in mind
To accommodate the unique requirements of complex creative and open-ended work, new crowdsourcing techniques consider global goals (rather than just local ones) by allowing workers to participate in how work is merged. For example, workers can combine the best contributions from multiple past workers [41] or repurpose old work for a new goal [14]. Other techniques help workers maintain global consistency: in classification tasks, context regarding the taxonomy developed so far is provided to workers as they arrive to complete tasks in order to allow workers to consider existing categories as they classify items [9, 3]. Context trees [38] recursively merge subparts of a long story to gather an emergent understanding of the larger plot; this strategy explicitly shifts from looking at low-level input to the larger story structure and vice-versa, but does not allow workers to modify the story or summary. Voting on how to keep work consistent and organizing high-level ideas prior to work can also help workers think about work from a global standpoint [13].

Another body of past work focuses on allowing workers to self-coordinate. In these, tasks are generated—either automatically or by a human leader—according to overall requirements and are made available for workers to take. The crowdware paradigm [42] proposes use of a shared todo list of collaboration tasks to solve global constraints in tasks that are hard to decompose (such as planning travel). Apparition [23] features a self-coordinating crowd, but workers do not directly reflect on their own organizational strategies, nor can they alter the directions laid out by the designer. The MicroWriter [36] similarly focuses on scaffolding direct, co-located collaboration between non-crowd groups, providing a shared space to generate, organize, and act on ideas. This shared space allowed pre-existing groups to benefit from a bottom-up approach of building ideas into written paragraphs through microtasks. Mechanical Novel explores a complementary top-down approach where workers first select a goal based on previous work in order to minimize the effort required to coordinate an unaffiliated crowd.

In other work, leaders and collaborators work together more directly; in animation production [27], writing [17, 29], and ideation [8], leaders distribute responsibility by generating tasks around which collaborators focus their efforts. However, in these systems, individual changes are requested and vetted by the same person, and contributors are often able to directly communicate with the leader. Instead, Mechanical Novel looks at how crowd workers can iteratively collaborate with each other, and introduces a technique for iterating on a central goal without a central creative authority.

Mechanical Novel expands on past research by exploring how evaluating lower-level work against global goals can help crowd workers generate globally consistent output. In addition, workers choose goals themselves. Based on this, we hypothesize that allowing crowdworkers to influence both high-level and low-level work may help workers converge on a common creative direction. By allowing them to revise, we open opportunities for workers to challenge and change the constraints of their work when appropriate.

MECHANICAL NOVEL
To enable crowds to manage high-level interdependencies as they collaborate on complex work, we introduce a crowdsourcing technique consisting of two phases. First, crowd workers reflect to brainstorm and choose a high-level goal to pursue. Second, workers revise their work to achieve this goal by decomposing that goal into specific tasks. This process loops to continually improve previous work. We test our technique for crowd reflection and revision in a system for collaborative fiction writing called Mechanical Novel. In this section, we describe the workflow (Figure 1) that guides the crowd through a collaborative revision process.

Designing workflows based on expert practice
Our technique takes inspiration from expert creative practice. Experts do indeed break down their work into smaller parts, but not independent tasks: rather, they continuously reflect on their work and use that reflection to revise their goals and decide what to do next [11]. An author, for example, does not finish a story after simply linearly filling in a plot outline—they instead write and rewrite while continually reflecting on what their vision is and how to achieve it [34]. Similar processes occur across many creative domains such as art, architecture, and writing [33, 11, 2]. This process, looping between reflecting on progress to identify a goal and revising based on that goal, allows experts to “converse” with their work [33] and evaluate options by trying them out [31]. However, because crowd workers are typically not domain experts, this strategy needs to take the form of microtasks in order to use it in crowdsourcing systems. Our intent here is not to reduce storytelling to an impassive and mechanic series of steps; in fact, we chose storytelling as an example domain to help us develop the technique we describe in this paper precisely because it requires a flexible process that can respond to flashes of inspiration and emotional sensibility. Designing for non-traditional work tasks (such as writing stories) may uncover new types of crowdsourcing and collaboration techniques that preserve the ability to respond to creative insight.

Initialization: creating a first draft
For Mechanical Novel to engage in reflection and revision, it must begin with a first draft. The first draft is authored using traditional iterative crowdsourcing strategies. Mechanical Novel initially takes a short prompt describing the overall concept of the story as input, such as “A young boy named
Figure 2. Workers critique a story, reflecting on what is working and not working in order to choose a goal for their work.

Figure 3. Workers select scenes to unlock for revision, suggesting how each scene should change to help achieve the goal.

Malcolm finds himself alone in a runaway hot air balloon and accidentally travels to a city in the sky. Based on this prompt, the crowd generates the first draft of a story that is six scenes long. 1 Scenes are the basic unit of writing work in Mechanical Novel; rather than allowing workers to edit any part of the text they like, the system restricts workers to editing within one scene during any task.

To do this, five workers each independently write a candidate for the text of a scene. Other workers then vote for the best candidate, and Mechanical Novel advances to the next scene. Scenes are written sequentially—from the first to the last—rather than in parallel, to aid workers in coordinating lower-level details such as character names, mood, or writing style. Though this sometimes results in chaotic stories that rapidly change direction, forcing sequentiality ensures that workers concretely define possible creative directions that later workers can choose from when deciding how to improve the story.

Reflect: choosing a high-level goal

At this point, the crowd has created a first draft of a story, which is likely rife with narrative inconsistencies. To set a high-level goal for subsequent work, we break down the task of reflection into two steps. First, to generate possible goals to pick from, a new set of workers reads the current version of the story and then generates five critiques using the I like—I wish—what if method (Figure 2) [10]. Using this method, workers each write one sentence about what they liked about the story (“I like…”), one sentence about what they wish were different about the story (“I wish…”), and one sentence suggesting a concrete change to the story that would make it better (“What if…?").

Then, to determine which goal is most pressing or interesting to pursue, other workers then vote for the critique they agree with most. The “what if?” with the most votes becomes the chosen goal for later work (e.g., “What if the story ended with Malcolm learning a lesson about the importance of family?”). In this way, workers identify a new goal for work by reacting to the problems present in the current draft.

Revise: translate goals into actionable tasks

The revision phase of work is divided into four steps. Workers first vote to indicate which of the story’s scenes they think must change in order to achieve the goal. Voting for more than one scene indicates that there are dependencies in the story that require multiple parts of the story to change at the same time. For each scene they vote for, workers also must write a short one-sentence suggestion for how that scene must change in order to achieve the goal (e.g., “Malcolm should apologize to his grandfather in this scene.”) to generate possible revisions to choose from.

Scenes that at least four (out of 10) workers vote for are then unlocked for editing. For each of the unlocked scenes, a new set of workers vote for the suggestion they think best represents how the scene should change. The suggestions with the highest votes for each of the unlocked scenes become tasks that direct how the story should change.

Mechanical Novel then asks workers to sequentially fix each unlocked scene, presenting to workers both the high-level goal as well as instructions for incorporating the suggestion into the scene. Fixing a scene involves two more tasks similar to those used to write the first draft; multiple workers propose

---

1This story length struck a balance between being long enough to make it difficult to coordinate work and short enough to complete in a reasonable amount of time on Mechanical Turk.
new versions of the scene based on the task’s instructions, then other workers vote for the version that best achieves the suggestion and the higher-level goal. This process is repeated across each unlocked scene. In this way, the high-level goal serves the purpose of restricting the space of possible contributions from workers.

At this point, workers continue to improve the story by returning to the reflection phase, reading the new version of the story and submitting another set of critiques. They then vote for a new high-level goal, split that goal into actionable tasks, and modify the story based on those tasks, resolving different problems with the story with each revision. Currently, story writing stops after a predetermined number of revision rounds, but in future work, Mechanical Novel could allow the crowd to decide when to end the story (e.g., through votes).

**EVALUATION**

Mechanical Novel hypothesizes that structuring work around reflecting and revising high-level goals can allow the crowd to collaborate on complex interdependent work such as fiction writing. In this section, we report on two evaluations exploring whether or not this technique resulted in higher quality stories. In sum, these evaluations find that Mechanical Novel produces stories that were overall preferred over those written using an iterative crowdsourcing strategy, and that it was especially effective at finding and fixing high-level plot issues.

Specifically, the first evaluation gauged how well Mechanical Novel could detect and fix known narrative issues in a series of benchmark stories. The second evaluation compared the quality of stories written by Mechanical Novel and a typical iterative (CrowdForge-style) workflow when given an open-ended story prompt.

Both studies consisted of two experimental conditions (Figure 6): the *Mechanical Novel* condition, where workers wrote stories by reflecting on a first draft to choose a goal and then revising text, and a *control* condition, where workers wrote stories by voting for which parts of the story to edit and made independent edits to the story’s text. The workflows in the Mechanical Novel and control conditions both included tasks where workers unlocked and edited scenes; the workflow for the Mechanical Novel condition included the additional step of reflecting to set a high-level goal.

Figure 6 shows the tasks workers did for each revision of stories in each study condition. All tasks were launched simultaneously on Amazon Mechanical Turk to United States workers with a task approval rating of 90% or higher. Tasks, including those used to generate first drafts, were estimated to take 2 to 8 minutes to complete. Because we wanted to prevent workers from doing tasks from different experimental conditions, we were unable to price different types of tasks individually; instead, we paid all workers based on the longest possible task (-priced at $0.85 each to achieve at an hourly wage of at least the federal minimum wage\(^2\) on average, in accordance with Mechanical Turk guidelines for academic requesters [39]). Workers who participated in our tasks were randomly assigned to one of the study conditions for the entirety of their interaction with the system.

**Benchmark Study**

Our first evaluation sought to measure Mechanical Novel’s performance on six benchmark stories with known narrative issues. This evaluation helps us understand the kinds of high level goals that Mechanical Novel can set and execute.

We ran the Mechanical Novel and control versions of the system for a single revision cycle over six pre-written benchmark task stories, resulting in 12 stories. Each of the benchmark stories were modified versions of a single short story written by an expert with over 10 years of fiction writing experience (including a crowdfunded, self-published children’s novel). Each modified version was changed by the expert to introduce one major problem each (Table 1). We chose both problems that can be fixed independently (such as fixing typos) as well as problems that span across the story (such as changing the way a character speaks) to get a better sense of Mechanical Novel’s strengths and weaknesses.

To get a sense of how often workers were able to find problems in the benchmark stories, we tracked the number of times workers correctly voted to change problematic scenes. Two of the authors, blind to condition, then coded edits made to each story in each condition to track how often workers were able to fix the correct problem for each condition ($\chi^2 = 0.93$). This was repeated three times for each story, so each story had three separate chances to fix errors. Workers were randomized into one of the six stories within each condition, and were only allowed to contribute to one of the repetitions.

**Results**

The benchmark study suggested that Mechanical Novel is effective at detecting high-level narrative problems (Table 2). Compared to the control condition, the Mechanical Novel condition resulted in significantly more Turkers correctly voting to change the problematic section of a story for the *abrupt ending* problem ($\chi^2(1) = 4.82, p < 0.05$) and tended towards correctly identifying problematic sections for the *extra characters* problem ($\chi^2(1) = 3.33, p = 0.068$) according to Chi-squared tests. The control condition, on the other hand, identified fewer problems correctly.

\(^2\)The federal minimum wage at the time of this writing was $7.25.
Problem Description

- **Abrupt ending**: The ending of the story is replaced with a sudden exclamation that the story was actually a dream all along.
- **Extra characters**: Dialogue and actions by unnecessary characters are added throughout the story.
- **Odd dialogue**: The main character, who is a child, is changed so that he speaks like an adult.
- **Point-of-view change**: The story changes from third-person to first-person narration halfway through.
- **Typos**: Grammar and spelling errors are introduced to some of the scenes in the story.
- **Tell, not show**: Character’s actions are replaced with descriptions of boring or unrealistic behavior (e.g. “Malcolm’s mother held a finger to her lips” v.s. “Malcolm’s mother told him to be quiet.”)

Table 1. The benchmark stories were modified versions of a short story created by an expert, each introducing a common storywriting problem.

| Problem       | Description                                                                 |
|---------------|-----------------------------------------------------------------------------|
| Abrupt ending | The ending of the story is replaced with a sudden exclamation that the story was actually a dream all along. |
| Extra characters | Dialogue and actions by unnecessary characters are added throughout the story.         |
| Odd dialogue  | The main character, who is a child, is changed so that he speaks like an adult.               |
| Point-of-view change | The story changes from third-person to first-person narration halfway through.                         |
| Typos         | Grammar and spelling errors are introduced to some of the scenes in the story. |
| Tell, not show | Character’s actions are replaced with descriptions of boring or unrealistic behavior (e.g. “Malcolm’s mother held a finger to her lips” v.s. “Malcolm’s mother told him to be quiet.”) |

Table 2. The total votes cast by workers choosing which sections of the benchmark stories to edit, as well as the number of votes correctly identifying problematic story sections.

| Problem       | Control Correct % | MNovel Correct % |
|---------------|--------------------|------------------|
| Abrupt ending | 30%                | 61%              |
| Extra characters | 43%             | 70%              |
| Odd dialogue  | 48%                | 67%              |
| POV change    | 40%                | 57%              |
| Typos         | 50%                | 9%               |
| Tell, not show | 23%                | 23%              |

Table 3. The total number of edits made by workers to benchmark stories, as well as the number of paragraphs that correctly corrected problematic story sections.

| Problem       | Control Correct % | MNovel Correct % |
|---------------|--------------------|------------------|
| Abrupt ending | 25%                | 67%              |
| Extra characters | 4%                | 0%               |
| Odd dialogue  | 18%                | 50%              |
| POV change    | 60%                | 50%              |
| Typos         | 33%                | 29%              |
| Tell, not show | 0%                 | 25%              |

Figure 6. The tasks launched on Mechanical Turk for each experimental condition.

Mechanical Novel

- Write First Draft (5 writing & 10 voting tasks per scene)
- Critique story (5 tasks)
- Vote for goal (10 tasks)
- Choose scenes to edit (10 tasks)
- Suggest edits based on goal (10 tasks)
- Edit scenes (5 writing & 10 voting tasks per scene)

Control

- (5 writing & 10 voting tasks per scene)
- Repeat

Story Writing Study

After establishing that Mechanical Novel allows workers to collaborate to identify high-level goals and to execute them, we wanted to understand how well Mechanical Novel would perform not just in terms of correcting high-level errors but in terms of developing stories from scratch compared to a system representing the state of the art.
Lastly, we conducted a grounded theory analysis of how stories changed in each condition by coding the types of changes made in each condition as well as the feedback we received from the crowd workers who worked on writing tasks. Two of the authors, blind to condition, also independently coded each dataset according to emergent themes and resolved conflicts through discussion (paragraph edits: $\kappa = 0.74$; critiques: $\kappa = 0.86$; task feedback: $\kappa = 0.61$).

### Results

Five stories were written for the Mechanical Novel and control conditions, resulting in 10 stories total written by crowd-workers on Mechanical Turk. Stories took an average of 11.38 days ($SD = 1.42$) to complete (based on the timestamps of the first and last interactions with the story). Stories were generated through a total of 428 Mechanical Turk tasks completed by an average of 224.5 unique workers per story ($SD = 15.63$).

When rating the final stories overall, workers indicated they liked Mechanical Novel stories better (133 votes for Mechanical Novel v.s. 82 votes for the control workflow; $X^2(1) = 12.098, p < 0.01$), according to a Chi-squared test.

**Mechanical Novel stories developed story structure.** Readers rated Mechanical Novel stories as having significantly more complete plots ($X^2(1) = 28.698, p < 0.01$)—that is, readers indicated they viewed Mechanical Novel stories as having more of a complete story arc with a beginning, middle, and end compared to their control version counterparts. Readers also rated Mechanical Novel stories as having significantly more original story premises ($X^2(1) = 17.635, p < 0.01$). Considering that Mechanical Novel and control stories for the same story prompt started from the same first drafts, this may indicate that revising stories using high-level goals allowed story ideas to develop in more interesting ways, or that Mechanical Novel stories were more successful at maintaining the story idea established in the first draft.

The Blue Elephant story is an example of how Mechanical Novel was able to generate a more complete story arc. In the first draft of the story, the main character (a young girl) realizes her stuffed elephant is gone, looks all over it, and is finally reunited with it after finding that it has come to life. In the control condition, workers attempted to motivate the main character’s actions by establishing that the young girl considers her elephant her best friend. They also add a reason for the elephant’s disappearance by having the elephant say he had gone on an adventure.

Mechanical Novel workers, in contrast, revised the story’s beginning to include a description of how Kaley received the elephant from her grandmother, which was the same doll her recently deceased mother had when she was a little girl. Workers called back to this backstory in the ending of the story, which reveals that Kaley’s love for her grandmother is what brought the Blue Elephant to life, threading a specific theme through the whole story and tying it together.

**Mechanical Novel focused on story over proofreading.** Readers rated the control stories as having fewer grammar and spelling mistakes ($X^2(1) = 10.868, p < 0.01$), indicating that the workers in the control condition seemed to focus more on low-level edits and proofreading. In contrast, readers rated Mechanical Novel stories as having better use of

---

| Title                  | Prompt                                                                 |
|------------------------|-------------------------------------------------------------------------|
| The Blue Elephant      | Kaley is a girl who spends all her time with an old Blue Elephant doll that was passed down from her grandmother. One day, it disappears. |
| John Dough             | A cutthroat businessman realizes that he’s dead and has ended up in heaven, but he has unfinished business...                                      |
| The Hot Air Balloon    | A young boy named Malcolm finds himself alone in a runaway hot air balloon and accidentally travels to a city in the sky.                           |
| The High-Waisted Shorts| Emelia and her high school friends hang out on a normal day, when suddenly, she sees the ghost of a girl wearing beautiful flower-print high-waisted shorts. |
| Number 16              | A serial killer has been monitoring his next victim’s movements for months. She is a loner and the perfect target. One day she disappears and nobody notices but him. |

Table 4. Each study condition included five stories, each based on the prompts above. “Number 16” was adapted from Reddit’s /r/writingprompts.
imagery and description ($X^2(1) = 56.542, p < 0.01$) and as having writing styles that better matched each story idea ($X^2(1) = 23.447, p < 0.01$). The final versions of Mechanical Novel stories were also significantly longer than the final versions of the control stories ($t(4.77) = 3.65, p < 0.05$), with the Mechanical Novel stories having an average of 1010.6 ($SD = 226.15$) words, while the final versions of the control stories were an average of 623.8 ($SD = 70.47$) words long.

In sum, Mechanical Novel stories seemed to focus on fleshing out the story itself and how it was told, rather than focusing on local fixes such as missing punctuation or awkward sounding sentences. This is corroborated by the analysis of types of edits that workers made to each story (Table 6). We found that workers in the Mechanical Novel condition made significantly more edits that had to do with expanding on descriptions of characters and how they would act ($X^2(1) = 12.49, p < 0.01$), while workers in the control condition trended towards more edits related to fixing grammar and spelling ($X^2(1) = 3.202, p = 0.074$) and completely rewrote paragraphs significantly more often ($X^2(1) = 3.95, p < 0.05$). Table 7 also shows that Mechanical Novel workers favored high-level goals that improved high-level flow throughout the story over low-level goals (such as correcting spelling and grammar) and goals that would substantially change the story’s concept (such as reordering paragraphs).

An example of this can be seen when comparing the Mechanical Novel and control versions of the John Dough story. The control story starts out with a straightforward description of the character’s surroundings:

John Dough slowly awoke from a foggy haze. He sat up and immediately felt a searing pain shoot through the left side of his body.

Table 5. The questions asked to workers who compared the control and Mechanical Novel stories for each story prompt, as well as the number of workers who voted for the Control story or the MNovel story for each question. ($* = p < 0.05$)

| Category     | Question                                                                 | Control Votes | MNovel Votes |
|--------------|--------------------------------------------------------------------------|---------------|--------------|
| Imagery      | Which story uses better imagery and description? A story with good imagery has description that is memorable and makes it easier to imagine what is happening in the story. | 52            | 162*         |
| Coherency    | Which story is more coherent? A coherent story has details that are consistent. The story makes sense and doesn’t meander or jump around without explanation. | 98            | 115          |
| Plot         | Which story has a more complete plot? A complete plot has a beginning, middle, and end, with a conflict that arises and is resolved by the end of the story. | 67            | 145*         |
| Originality  | Which story is more original? An original story has a clear, interesting story premise. | 75            | 136*         |
| Style        | Which story better uses writing style to enhance the telling of the story? A story with good writing style chooses a voice and tone that makes sense given the story’s content and contributes to the telling of the story. | 72            | 143*         |
| Technical    | Which story has less grammar and spelling mistakes? | 130*         | 82           |
| Overall      | Which story did you like better, overall? | 82            | 133*         |

Table 6. Workers in the Mechanical Novel condition were especially likely to expand characters, while workers in the control condition were more likely to rewrite a scene from scratch. ($* = p < 0.05$)

| Edit Type         | Control | MNovel |
|-------------------|---------|--------|
| Expand characters | 3 6%*   | 14 37%*|
| Improve flow      | 4 7%    | 8 21%  |
| Add to plot       | 11 20%  | 5 13%  |
| Clarify or cut text | 10 19% | 5 13%  |
| Add story background | 2 4% | 2 5%   |
| Change story’s tone | 1 2% | 2 5%   |
| Rewrite scene     | 10 19%* | 1 3%*  |
| Correct technical issues | 9 17% | 1 3%   |
| Emphasize story’s moral | 4 7% | 0 0%   |

Table 7. Categories of edit types made by workers in the Mechanical Novel condition.

“Where am I?” he wondered out loud. John did not recognize the room he was in. Everything was white and pristine...white walls, white carpet, white couch and white table, and bright white lights. There was no window, and only a single door at the other end of the room.

—control condition, John Dough

The Mechanical Novel story, however, uses first-person voice to create vivid imagery of the main character’s thoughts and feelings as they wake up in an unfamiliar place:

I awoke with a start, sitting up abruptly. There was a searing pain shooting through my body.

“Where am I?” I thought to myself.

I didn’t recognize my surroundings. Everything was white and pristine; white walls, white carpet, white couch, and white table. No windows, a single door across the room... but somehow the room was intensely bright. Strange.

—Mechanical Novel condition, John Dough

John Dough slowly awoke from a foggy haze. He sat up and immediately felt a searing pain shoot through the left side of his body.
Novel stories were created by unique workers, with an average of 7.8 accepted changes per Mechanical Novel story ($SD = 1.48$) by an average of 7.4 unique workers ($SD = 1.34$). Revisions to control stories were distributed among workers similarly, with an average of 9.4 accepted changes per control story ($SD = 1.52$) by an average of 8.8 unique workers ($SD = 1.92$). There was no significant difference between study conditions in the number of unique workers whose revisions were accepted ($t(7.2) = 1.33, n.s.$). In addition, out of the Mechanical Novel workers who participated in more than one task, 62.2% participated in both reflect and revision phases of a story. Out of Mechanical Novel workers who completed at least 3 tasks, 92.2% did at least 2 different types of tasks and 70.6% did at least 3 different types of tasks. In other words, it was not the case that a few skilled workers were dominating story-writing tasks in Mechanical Novel.

Both conditions struggled with coherency. There was no significant difference between the control and Mechanical Novel stories in terms of how coherent they were perceived to be ($X^2(1) = 1.357, n.s.$)—that is, all stories were seen as lacking consistency in details (for example, in The Blue Elephant, workers did not resolve whether it was Kaley’s mother or grandmother who had passed away). In addition, in their feedback, there was no significant difference between conditions on how often workers expressed frustration with having to struggle against earlier or later parts of the story ($X^2(1) = 0.009, n.s.$):

That the person who wrote the paragraph before mine paid no attention to pacing and didn’t seem to know much about hot air balloons. The “accidentally knocked unconscious” cliche was a bit annoying...

—Worker, control condition, The Hot Air Balloon

### DISCUSSION

Through an analysis of how the crowd wrote stories through Mechanical Novel, we found that techniques for setting high-level goals—inspired by expert writers’ process—helped the crowd produce stories with stronger narrative arcs and description compared to stories written using a traditional crowdsourcing workflow.

### Enabling flexibility and encouraging diversity

We also found that Mechanical Novel spread work across many unique workers, rather than allowing a few skilled workers to dominate the creative process. This indicates that the reflect and revise technique we use in this paper provides a steady source of fresh perspectives on a complex task where creative exploration is necessary. The diversity of perspectives that this technique affords may expand the types of work crowdsourcing can support. For example, citizen journalism is recognized for its ability to disseminate news faster and with wider reach than mainstream news organizations. At the same time, much like crowdsourcing, it faces criticisms stemming from its decentralized nature; reports by citizen journalists are difficult to regulate and may not adhere to standards of quality, trustworthiness, objectivity, and ethics. While a professional journalist could help solve these problems, the...
presence of an expert also negates the value of citizen journalism as an alternative source of timely information. Enabling a crowd of decentralized contributors to revise and reflect together on the news they produce may preserve the ability to quickly propagate information while keeping each others’ facts and biases in check through brainstorming and voting for a common high-level goal. In addition, the ability to continually revise and act on new goals may allow crowds to work together in generating stories around events such as natural disasters where centralized information is unavailable.

Reflecting and revising on work is also medium-agnostic and can be implemented as part of a crowdsourcing system regardless of the actual work task at hand. With Mechanical Novel, we found that workers submitted and voted for critiques and edits appropriate for story writing (such as those that focused on plot and character) without having the system specify desired input from the crowd. For this reason, one could imagine that the crowd could use this technique to flexibly support work that moves through different stages of production. Reflections on the script for a crowdsourced film, for example, could lead to revisions of a storyboard or casting choices. Then, the actual task of creating the film could be supported through existing crowdsourcing strategies and interfaces (e.g., [20]).

However, we also found that Mechanical Novel performed less well than the control system when it came to low-level work (such as correcting grammar and spelling errors). This may mean that reflecting and revising could be used in a complementary way with existing crowdsourcing patterns; for example, find-fix-verify [5] could be used to refine the stories that Mechanical Novel generates.

**Going beyond short stories**

At the same time, Mechanical Novel is currently limited by its assumption that the short story being generated is small enough to fit in the working memory of each worker. That is, a worker has to be able to read the whole story and make a critique in order to select a high-level goal for subsequent work. In addition, workers currently must be able to look through the entire story to flag which parts of the story must change in order to achieve the high-level goal. For the purposes of exploring the approach of decomposing crowdsourced creative work based on a goal selected by the crowd, we deliberately limited the length of each story so that it is possible for each worker to familiarize themselves with the story in a short amount of time.

How might this approach be used to generate a larger work? One strategy may be to apply the Mechanical Novel approach recursively, where workers could collaborate on the high-level structure of a story, then dynamically expand on individual chapters or narrative acts. Another strategy may be to make use of a working memory space for the crowd as seen in past work [24, 42] to further help direct work by letting future workers know the creative intent of past workers.

**Designing collaboration around reflection and revision**

Why does the reflect and revise technique work? Too much structure can undesirably limit the work that crowd workers do. An early version of Mechanical Novel allowed crowd workers to set high-level goals by having them brainstorm and vote on an outline, much like Crowdforge [19]. Our intent here was to allow workers to concentrate on brainstorming the bigger picture without having to worry about the details of how the story would actually be written. However, we found that workers would work within the outline far too strictly (similar to worker behavior seen in other highly-structured crowd systems [21]). This made it hard for workers to explore a wide range of possible creative directions inspired by the story outline; they would not change much from the initial outline that was selected. This may be because crowd workers may err on the side of caution when told to make changes that may or may not be correct in order to avoid having their work rejected.

Instead, we had to design a way for workers to concretely explore possible creative directions. At first, we tried asking workers to brainstorm a theme or moral for the story that would ground later work. Though workers were generally able to select a reasonable theme to guide the next revision of a story, they had difficulty translating such an abstract high-level idea into concrete changes. Instead, critiques provided a way for workers to think about high-level changes in terms of what they wanted to story to specifically look like after revision took place. However, this means that the reflect and revise technique only works to the extent that non-experts can make evaluations. For example, the crowd may be able to select reasonable goals for changing the structure and flow of a research paper, but are less likely to assess a research paper in terms of how it compares to existing literature. Techniques such as scaffolding feedback [40] could help support the reflect phase of work in more specialized domains such as science or design.

Lastly, Mechanical Novel was designed around the constraints of Mechanical Turk, which rewards crowd workers for quickness and punishes workers for subpar work quality. A new kind of marketplace—perhaps one that encourages slower, thoughtful work or risky brainstorming—may better support the type of creative work described in this paper. In future work, experiments that probe into the relative difficulty of reflection compared to revision may help define the optimal incentive scheme such a market should provide. For example, revision may benefit from thoughtful and careful work while reflection may work best when workers are asked to make snap decisions (or vice versa); this, in turn, may require different reward systems (such as rewarding based on quantity versus quality).

**CONCLUSION**

In this paper, we enabled crowds to collaborate on complex creative work through a technique where the crowd reflects on their work and translates those reflections into concrete revisions of the work. When crowdwork is structured around reflection and revision, workers can identify and execute high-level goals even when work cannot be easily split into independent tasks. This approach allowed workers to detect and fix high-level storytelling problems and resulted in higher quality stories than those written using a traditional crowd-
Kaley started her search for the Blue Elephant. First, she searched her room looking in every nook and cranny. No Blue Elephant. After a very long day of searching and not finding Blue Elephant, Kaley started to cry. Kaley’s mother had an idea... She gave Kaley some peanuts to put out to help catch Blue Elephant. Finally Kaley fell asleep for the night. When she woke up in the morning, Blue Elephant was in her bed with a stash of peanuts.

The girl called for the elephant as loudly as she could. He must have heard her, his big ears make it possible to hear from miles away. If he were trapped or something he would surely be able to send a reply with his giant trunk. She wondered where he could be and wandered down the road calling loudly for him. Every few steps she would sit still and listen for him. Then, she thought she heard a muffled reply and put her ear to the ground. She felt the soft thump of an elephant from a far away distance.

Sure enough, Blue was floating gleefully in the pool spraying water triumphantly from his trunk. Kaley could scarcely believe her eyes, but the glee of her imagination took hold and she yelled in joy.

“Blue! Blue! Is that you?”

At the sound of her voice, the little elephant turned his trunk and blew water all over her. Leaving her soaking wet and giggling at her silly little friend.

“How did this happen?” asked Kaley. The blue elephant was delighted to answer her question. “You see, Kaley, it was through your love and adoration that I was able to come to life! If it weren’t for you I wouldn’t be here. Remember that wish you made the day before since? Well, it came true! The spirit of your grandmother lives on, in me. She wanted nothing but for you to be happy. Because of your love I’m here and will answer anything you ask” Shocked, Kaley took a step back and assessed the situation. “Well, I suppose this wasn’t such a bad wish!” She thought about what she would ask but really all she wanted was to tell her grandma that she missed her “I miss you grandma, you were gone too soon...” “ Your grandmother would be happy to hear that Kaley and please tell your mother that she loved her no matter how much things turned out”. “ Elephant? Are you going to stay?” “ I’m afraid not. Grandma’s spirit has given me only a temporary time with you and it’s just about to expire” Just like that the dol started to grow and landed in Kaley’s hand. Kaley hugged the doll. A doll that she will forever cherish.

REFERENCES
1. Foldingstory. http://foldingstory.com/.
2. Alexander, C., Ishikawa, S., and Silverstein, M. Pattern languages. Center for Environmental Structure 2 (1977).
3. André, P., Kittur, A., and Dow, S. P. Crowd synthesis: Extracting categories and clusters from complex data. In Proc. CSCW, CSCW ’14, ACM (New York, NY, USA, 2014), 989–998.
4. Bayles, D., Orland, T., and Morey, A. Art & fear. Tantor Media, Incorporated, 2012.
5. Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., and Panovich, K. Soylent: A word processor with a crowd inside. In Proc. UIST, UIST ’10, ACM (New York, NY, USA, 2010), 313–322.
6. Bigham, J. P., Bernstein, M. S., and Adar, E. Human-computer interaction and collective intelligence. In Collective Intelligence Handbook. MIT Press, 2015.
7. Burroway, J. Imaginative writing: The elements of craft. Longman, 2003.
8. Chan, J., Dang, S., Kremer, P., Guo, L., and Dow, S. Ideagens: A social ideation system for guided crowd brainstorming. In Second AAAI Conference on Human Computation and Crowdsourcing (2014).
39. WeAreDynamo. Guidelines for academic requesters. http://wiki.wearedynamo.org/index.php/Guidelines_for_Academic_Requesters. [Online; accessed 12-May-2015].

40. Xu, A., Huang, S.-W., and Bailey, B. Voyant: Generating structured feedback on visual designs using a crowd of non-experts. In Proc. CSCW, ACM (New York, NY, USA, 2014), 1433–1444.

41. Yu, L., and Nickerson, J. V. Cooks or cobblers?: Crowd creativity through combination. In Proc. CHI, ACM (New York, NY, USA, 2011), 1393–1402.

42. Zhang, H., Law, E., Miller, R., Gajos, K., Parkes, D., and Horvitz, E. Human computation tasks with global constraints. In Proc. CHI, CHI ’12, ACM (New York, NY, USA, 2012), 217–226.